

**Optimasi *Travelling Salesman Problem* Pada Angkutan  
Sekolah Menggunakan Algoritme Genetika  
(Studi Kasus : Sekolah MI Salafiyah Kasim Blitar)**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Ivarianti Sihaloho  
NIM: 145150201111130



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG

## PENGESAHAN

OPTIMASI *TRAVELLING SALESMAN PROBLEM* PADA ANGKUTAN SEKOLAH  
MENGUNAKAN ALGORITME GENETIKA  
(STUDI KASUS: SEKOLAH MI SALAFIYAH KASIM BLITAR)

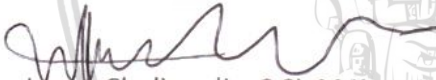
SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Ivarianti Sihalohe  
NIM: 145150201111130

Skripsi ini telah diuji dan dinyatakan lulus pada  
02 Agustus 2018  
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

  
Imam Cholissodin, S.Si, M.Kom  
NIK: 2012018507191001

Dosen Pembimbing II

  
Tibyani, S.T, M.T  
NIP: 196911011995121002

Mengetahui  
Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D  
NIP: 197105182003121001

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 02 Agustus 2018



Ivarianti Sihaloho

NIM: 145150201111130

## KATA PENGANTAR

Puji dan syukur saya ucapkan kepada Tuhan Yesus Kristus, karena berkat dan anugerah-Nya, saya dapat menyelesaikan skripsi ini. Skripsi ini ditulis sebagai salah satu syarat kelulusan dari Program Sarjana Ilmu Komputer Universitas Brawijaya. Saya mengaku bahwa dalam penulisan hingga sampai menyelesaikan skripsi ini terdapat banyak pihak yang membantu dan memberikan dukungan kepada saya. Oleh karena itu, saya ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Papa dan Mama tercinta, Mangatur Sihaloho dan Denni Sitanggang yang selalu mendoakan, memberi dukungan baik secara moral maupun materil, dan juga motivasi terbesar saya untuk segera menyelesaikan skripsi ini. Skripsi ini saya persembahkan untuk Papa dan Mama.
2. Kakak Vales Fitri Lidia Sihaloho dan Abang Paolotoma Sihaloho yang selalu memberikan semangat, motivasi dan percaya bahwa saya akan menyelesaikan skripsi ini tepat pada waktunya, disamping setiap aktivitas dan kesibukan yang saya tekuni di luar kampus.
3. Bapak Imam Cholissodin S.Si, M.Kom selaku dosen pembimbing satu saya yang sudah memberikan waktu, tenaga, pikiran serta kesabaran dalam membimbing dan mengarahkan saya dalam pengerjaan skripsi ini.
4. Bapak Tibyani, S.T,M.T selaku dosen pembimbing kedua saya yang sudah memberikan waktu, tenaga serta kesabaran dalam membimbing dan mengarahkan saya dalam pengerjaan skripsi ini.
5. Pihak sekolah MI Salafiyah Kasim Blitar yang sudah memberikan waktunya kepada saya untuk dapat ditemui dan terima kasih sudah mengijinkan saya untuk melakukan penelitian disana.
6. Mbak Maulani Camaliah M selaku kakak tingkat dan kakak kost saya yang sudah memberikan waktu, tenaga, pikiran dan membantu saya selama pengerjaan skripsi.
7. Junaty Silveria Sitanggang, Janter Simarmata, Anisah Denis, Rilinka Manullang, Nella Sarah, dan Octavia Hutasoit, atas dukungan, motivasi, doa dan saran yang sudah kalian berikan kepada saya sehingga skripsi ini dapat terselesaikan tepat waktu.

Malang, 02 Agustus 2018

Penulis

ivasihaloho@gmail.com

## ABSTRAK

**Ivarianti Sihaloho, Optimasi *Travelling Salesman Problem* Pada Angkutan Sekolah Menggunakan Algoritme Genetika (Studi Kasus: Sekolah MI Salafiyah Kasim Blitar)**

**Pembimbing: Imam Cholissodin, S.Si, M.Kom dan Tibyani, S.T, M.T**

Modernisasi yang berkembang masa ini berdampak pada berbagai aspek dalam kehidupan masyarakat salah satunya adalah pada bidang pendidikan yang mulai mengembangkan fasilitas bagi para siswanya yaitu berupa angkutan sekolah. Salah satu sekolah yang memfasilitasi angkutan sekolah adalah sekolah MI Salafiyah Kasim yang berada di Blitar. Dalam pelaksanaannya masih terdapat berbagai kendala seperti supir yang mengedepankan pengalaman pribadi, siswa yang diantar pada setiap harinya tidak selalu sama, keterlambatan supir saat mengantar hingga biaya operasional yang belum stabil. Dalam mengatasi hal tersebut, penelitian ini menggunakan algoritme genetika untuk melakukan optimasi *route* pengantaran. Hasil yang didapat dengan melakukan perbandingan antara data aktual yang dilalui supir sekolah dan data hasil rekomendasi sistem sejumlah 3 hari. Hasil optimasi dari sistem yang pada penelitian ini menghasilkan optimasi 5,5 km (19,78%) pada kloter pagi dan 17,17 km (36,30%) pada koter siang. Berdasarkan hasil tersebut dapat diketahui bahwa sistem dapat melakukan optimasi dengan baik sehingga diharapkan dapat meningkatkan kualitas pelayanan pengantaran siswa pada sekolah MI Salafiyah Kasim.

**Kata kunci:** rute, angkutan sekolah, optimasi, *travelling salesman problem* (TSP), algoritme genetika



## ABSTRACT

**Ivarianti Sihaloho, *Traveling Salesman Problem On School Transport With Genetic Algorithm (Case Study: Salafiyah Kasim Blitar School)***

**Advisor: Imam Cholissodin, S.Si, M.Kom and Tibyani, S.T, M.T**

Modernisation of educational programmes in school are now fundamentally dependent on the information technology, especially for the school bus service in MI Salafiyah Kasim Elementary School, which is located in Blitar. In this case, there are still many problems like sometimes the school bus drivers come late to pick up the students because of their personal business, there are so many different students that must be picked up every single day, and the unstable Accruad Expenses. So, this research uses the genetic algorithm in order to manage the optimization of the arrival-departure schedules. The results of the optimization can be obtained by comparing the actual data of the arrival-departure schedules from the drivers of the school bus, that consist the data samples within 3 days of the arrival and departure schedules. The optimization result in this reaserch is about 5.5 km (19.78%) in the morning departure and 17.17 km (36.30%) in the afternoon arrival.

**Keywords:** routes, school transportation, optimization, traveling salesman problem (TSP), genetic algorithm

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT .....	vi
DAFTAR ISI .....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xii
DAFTAR PERSAMAAN.....	xiv
DAFTAR KODE PROGRAM .....	xv
DAFTAR LAMPIRAN .....	xvi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan .....	3
1.4 Manfaat.....	3
1.5 Batasan Masalah .....	3
1.6 Sistematika Pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN .....	6
2.1 Kajian Pustaka .....	6
2.2 MI Salafiyah Kasim .....	9
2.3 <i>Travelling Salesman Problem (TSP)</i> .....	10
2.4 Optimasi.....	11
2.5 Algoritme Genetika .....	11
2.5.1 Inisialisasi (Populasi Awal).....	11
2.5.2 Reproduksi .....	12
2.5.3 Evaluasi.....	13
2.5.4 Seleksi.....	13
BAB 3 METODOLOGI .....	14
3.1 Tipe Penelitian .....	14
3.2 Strategi Penelitian.....	14

3.3 Partisipan Penelitian .....	14
3.4 Lokasi Penelitian .....	14
3.5 Teknik Pengumpulan Data .....	14
3.6 Implementasi Algoritme .....	15
3.7 Teknik Analisis Data .....	15
3.8 Jadwal Penelitian .....	15
BAB 4 PERANCANGAN .....	16
4.1 Formulasi Permasalahan.....	16
4.2 Proses Perancangan Algoritme Genetika .....	17
4.3 Tahapan-tahapan Algoritme Genetika (GA) .....	18
4.3.1 Inisialisasi Populasi Awal .....	18
4.3.2 <i>Crossover</i> .....	22
4.3.3 Mutasi .....	25
4.3.4 Evaluasi.....	27
4.3.5 Seleksi.....	29
4.3.6 Perhitungan Nilai <i>Fitness</i> .....	31
4.4 Perhitungan Manualisasi .....	33
4.4.1 Inisialisasi Parameter Algoritme Genetika .....	33
4.4.2 Representasi Kromosom .....	33
4.4.3 Inisialisasi Populasi Awal .....	34
4.4.4 <i>Crossover</i> .....	36
4.4.5 Mutasi .....	38
4.4.6 Evaluasi.....	40
4.4.7 Seleksi.....	43
4.5 Perancangan <i>User Interface</i> .....	44
4.5.1 Perancangan <i>User Interface</i> Halaman Utama.....	44
4.5.2 Perancangan <i>User Interface</i> Halaman Proses Algoritme Genetika .....	45
4.5.3 Perancangan <i>User Interface</i> Halaman Hasil Rute .....	46
4.5.4 Pengujian Ukuran Populasi .....	47
4.5.5 Pengujian Kombinasi Nilai <i>Crossover Rate</i> dan <i>Mutation Rate</i> .....	48
4.5.6 Pengujian Uji Konvergensi .....	48



BAB 5 IMPLEMENTASI .....	50
5.1 Spesifikasi Sistem .....	50
5.1.1 Spesifikasi Perangkat Keras .....	50
5.1.2 Spesifikasi Perangkat Lunak .....	50
5.2 Implementasi Program .....	50
5.2.1 Implementasi Proses Inisialisasi Populasi Awal .....	51
5.2.2 Implementasi Proses <i>Crossover</i> .....	52
5.2.3 Implementasi Proses Mutasi .....	55
5.2.4 Implementasi Proses Evaluasi .....	57
5.2.5 Implementasi Proses Seleksi .....	58
5.3 Implementasi <i>User Interface</i> .....	59
5.3.1 Implementasi <i>User Interface</i> Halaman Utama .....	59
5.3.2 Implementasi <i>User Interface</i> Halaman Proses Algoritme Genetika .....	60
BAB 6 PENGUJIAN dan ANALISIS .....	68
6.1 Pengujian Ukuran Populasi .....	68
6.2 Pengujian Kombinasi Nilai <i>Crossover Rate</i> (Cr) dan <i>Mutation     Rate</i> (Mr) .....	71
6.3 Pengujian Konvergensi .....	75
6.4 Hasil Analisis Global .....	77
BAB 7 PENUTUP .....	81
7.1 Kesimpulan .....	81
7.2 Saran .....	81
DAFTAR PUSTAKA .....	82
Lampiran A DATA SISWA .....	83
Lampiran B DATA MATRIKS JARAK SISWA .....	87
Lampiran C HASIL PENGUJIAN .....	90

## DAFTAR TABEL

Tabel 2.1 Kajian Pustaka .....	6
Tabel 4.1 Data Siswa Kloter Pagi .....	16
Tabel 4.2 Data Siswa Kloter Siswa .....	16
Tabel 4.3 Representasi Kromosom Pagi .....	34
Tabel 4.4 Representasi Kromosom Siang .....	34
Tabel 4.5 Inisialisasi Awal Pagi .....	35
Tabel 4.6 Inisialisasi Awal Siang .....	35
Tabel 4.7 <i>Crossover</i> Pagi .....	36
Tabel 4.8 <i>Crossover</i> Siang .....	37
Tabel 4.9 Hasil <i>Offspring</i> dari Kloter Pagi .....	37
Tabel 4.10 Hasil <i>Offspring</i> dari Kloter Siang .....	38
Tabel 4.11 Mutasi Pagi .....	38
Tabel 4.12 Mutasi Siang .....	39
Tabel 4.13 Hasil <i>Offspring</i> dari Kloter Pagi .....	40
Tabel 4.14 Hasil <i>Offspring</i> dari Kloter Siang .....	40
Tabel 4.15 Hasil Populasi Baru Kloter Pagi .....	41
Tabel 4.16 Hasil Populasi Baru Kloter Siang .....	41
Tabel 4.17 Hasil Perhitungan Nilai <i>Fitness</i> Kloter Pagi .....	42
Tabel 4.18 Hasil Perhitungan Nilai <i>Fitness</i> Kloter Siang .....	42
Tabel 4.19 Hasil Seleksi Kloter Pagi .....	43
Tabel 4.20 Hasil Seleksi Kloter Siang .....	44
Tabel 4. 21 Pengujian Ukuran Populasi .....	47
Tabel 4. 22 Pengujian Kombinasi Nilai <i>Cr</i> dan <i>Mr</i> .....	48
Tabel 4.23 Pengujian Uji Konvergensi .....	49
Tabel 5.1 Spesifikasi Perangkat Keras .....	50
Tabel 5.2 Spesifikasi Lunak .....	50
Tabel 6.1 Hasil Pengujian Ukuran Populasi Kloter Pagi .....	68
Tabel 6.2 Hasil Pengujian Ukuran Populasi Kloter Siang .....	69
Tabel 6.3 Penggabungan Hasil Pengujian Ukuran Populasi .....	70
Tabel 6.4 Hasil Pengujian Kombinasi Nilai <i>Cr</i> dan <i>Mr</i> Kloter Pagi .....	72
Tabel 6.5 Hasil Pengujian Kombinasi Nilai <i>Cr</i> dan <i>Mr</i> Kloter Siang .....	73

Tabel 6.6 Penggabungan Hasil Pengujian Kombinasi Nilai <i>Cr</i> dan <i>Mr</i> .....	74
Tabel 6.7 Hasil Pengujian Konvergen Kloter Pagi .....	75
Tabel 6.8 Hasil Pengujian Konvergen Kloter Siang.....	76
Tabel 6.9 Data Aktual Rute Pengantaran .....	78
Tabel 6.10 Data Rekomendasi Hasil Sistem .....	79
Tabel 6.11 Hasil Selisih .....	80



## DAFTAR GAMBAR

Gambar 2.1 (a) Tampilan Depan Gedung Sekolah, (b) Tampilan Samping Gedung Sekolah .....	9
Gambar 2.2 (a) Tampilan Depan Bus Sekolah, (b) Tampilan Dalam Bus Sekolah. ....	10
Gambar 2.3 Tahapan Algoritme Genetika .....	11
Gambar 2.4 Reproduksi <i>Crossover One Cut Point</i> .....	12
Gambar 2. 5 Reproduksi <i>Exchange Mutation</i> .....	13
Gambar 4.1 Diagram Alir Algoritme Genetika Pada Rute Angkutan Sekolah.....	18
Gambar 4.2 Diagram Alir Inisialisasi Awal.....	19
Gambar 4.3 Diagram Alir Individu.....	20
Gambar 4.4 Diagram Alir <i>Random</i> .....	21
Gambar 4.5 Diagram Alir setKromosom .....	21
Gambar 4.6 Diagram Alir <i>Crossover</i> .....	23
Gambar 4.7 Diagram Alir <i>One Cut Point Crossover</i> .....	24
Gambar 4.8 Diagram Alir Mutasi.....	25
Gambar 4.9 Diagram Alir <i>Exchange Mutation</i> .....	27
Gambar 4.10 Diagram Alir Evaluasi.....	29
Gambar 4.11 Diagram Alir Seleksi.....	31
Gambar 4.12 Diagram Alir Perhitungan Nilai <i>Fitness</i> .....	32
Gambar 4.13 Tampilan Halaman Utama.....	45
Gambar 4.14 Halaman Proses Algoritme Genetika .....	46
Gambar 4.15 Halaman Hasil Rute .....	47
Gambar 5.1 <i>User Interface</i> Halaman Utama.....	60
Gambar 5.2 <i>User Interface</i> Populasi Awal Kloter Pagi .....	61
Gambar 5.3 <i>User Interface</i> Populasi Awal Kloter Siang .....	61
Gambar 5.4 <i>User Interface Crossover</i> Kloter Pagi.....	62
Gambar 5.5 <i>User Interface Crossover</i> Kloter Siang .....	62
Gambar 5.6 <i>User Interface</i> Mutasi Kloter Pagi .....	63
Gambar 5.7 <i>User Interface</i> Mutasi Kloter Siang.....	63
Gambar 5.8 <i>User Interface</i> Mutasi Kloter Siang.....	63
Gambar 5.9 <i>User Interface</i> Evaluasi Kloter Siang.....	64
Gambar 5.10 <i>User Interface</i> Evaluasi Kloter Siang.....	64

Gambar 5.11 <i>User Interface</i> Seleksi Kloter Siang.....	65
Gambar 5.12 <i>User Interface</i> Seleksi Kloter Siang.....	65
Gambar 5.12 <i>User Interface</i> Individu Terbaik Kloter Pagi.....	66
Gambar 5.13 <i>User Interface</i> Individu Terbaik Kloter Siang.....	66
Gambar 5.15 <i>User Interface</i> Individu Terbaik Kloter Siang.....	66
Gambar 5.14 <i>User Interface</i> Hasil Rute Kloter Siang .....	67
Gambar 6.1 Grafik Hasil Pengujian Ukuran Populasi Kloter Pagi.....	69
Gambar 6.2 Grafik Hasil Pengujian Ukuran Populasi Kloter Siang.....	70
Gambar 6.3 Grafik Penggabungan Hasil Rata-rata <i>Fitness</i> Pengujian Ukuran Populasi .....	71
Gambar 6.4 Grafik Hasil Pengujian Kombinasi <i>Cr</i> dan <i>Mr</i> Kloter Pagi .....	72
Gambar 6.5 Grafik Hasil Pengujian Kombinasi <i>Cr</i> dan <i>Mr</i> Kloter Siang .....	74
Gambar 6.6 Grafik Penggabungan Rata-rata <i>Fitness</i> Kombinasi Nilai <i>Cr</i> dan <i>Mr</i> . ..	75
Gambar 6.7 Grafik Hasil Pengujian Konvergen Kloter Pagi.....	76
Gambar 6.8 Grafik Hasil Pengujian Konvergen Kloter Siang.....	77

## DAFTAR PERSAMAAN

Persamaan 2.1 .....	11
Persamaan 2.2 .....	13





## DAFTAR KODE PROGRAM

Kode Program 5.1 Implementasi Inisialisasi Awal .....	51
Kode Program 5.2 Implementasi Proses <i>Crossver</i> .....	52
Kode Program 5.3 Implementasi Method <i>Crossover</i> .....	54
Kode Program 5.4 Implementasi Proses Mutasi.....	55
Kode Program 5.5 Implementasi Method Mutasi .....	56
Kode Program 5.6 Implementasi Evaluasi .....	57
Kode Program 5.7 Implementasi Seleksi .....	58



## DAFTAR LAMPIRAN

### LAMPIRAN A DATA SISWA

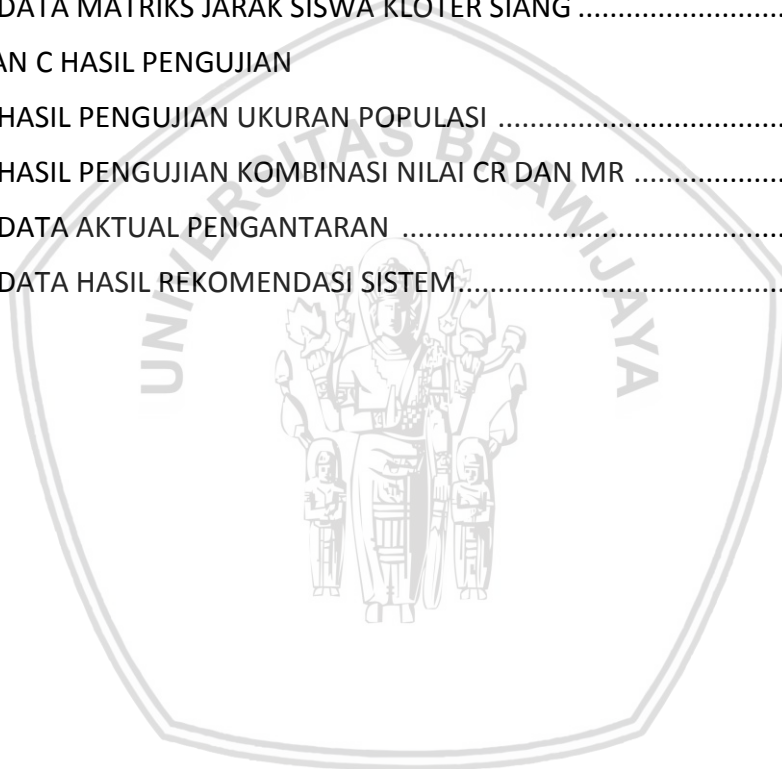
A.1 DATA SISWA KLOTER PAGI .....	83
A.2 DATA SISWA KLOTER SIANG .....	84
A.3 PELETAKAN TITIK ALAMAT SISWA KLOTER PAGI DI MAPS .....	85
A.4 PELETAKAN TITIK ALAMAT SISWA KLOTER SIANG DI MAPS .....	86

### LAMPIRAN B DATA MATRIKS JARAK SISWA

B.1 DATA MATRIKS JARAK SISWA KLOTER PAGI.....	87
B.2 DATA MATRIKS JARAK SISWA KLOTER SIANG .....	88

### LAMPIRAN C HASIL PENGUJIAN

C.1 HASIL PENGUJIAN UKURAN POPULASI .....	90
C.2 HASIL PENGUJIAN KOMBINASI NILAI CR DAN MR .....	92
C.3 DATA AKTUAL PENGANTARAN .....	94
C.4 DATA HASIL REKOMENDASI SISTEM.....	97



## BAB 1 PENDAHULUAN

Bab pendahuluan akan membahas tentang hal yang mendasari dari penelitian ini. Pada bab ini terdiri dari latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, dan sistematika pembahasan dari penelitian ini yang dijelaskan sebagai berikut.

### 1.1 Latar Belakang

Sekolah merupakan salah satu pusat pendidikan yang dibutuhkan oleh masyarakat untuk mendapatkan ilmu-ilmu bermanfaat yang bisa membuat orang-orang dapat berpikir semakin lebih maju (Daryanto, 2012). Ilmu-ilmu yang telah didapatkan akan menjadi dasar persiapan awal sebelum terjun ke dalam dunia pendidikan yang lebih tinggi lagi. Selain untuk mendapatkan ilmu, sekolah juga merupakan salah satu tempat untuk mengembangkan bakat, keterampilan yang dimiliki oleh masing-masing siswa dan di sekolah juga akan mengajarkan tentang bagaimana moral (budi pekerti) yang baik sebagai siswa yang berpendidikan. Pendidikan dapat dibagi menjadi beberapa kategori antara lain pendidikan SD, SMP, SMA dan Perguruan Tinggi. Pada umumnya, setiap sekolah menyediakan fasilitas antara lain meja belajar, ruang belajar, papan tulis, ruang laboratorium, komputer, ruang praktikum, fasilitas pelayanan dan lain-lain.

Salah satu fasilitas pelayanan yang disediakan oleh pihak sekolah untuk siswa adalah transportasi. Transportasi merupakan salah satu kegiatan yang dilakukan untuk mengangkut ataupun memindahkan barang dan manusia dari satu tempat ke tempat yang lainnya (Ahmad, 2005). Transportasi sangat dibutuhkan oleh masyarakat dan siswa sekolah misalnya angkutan umum, kendaraan motor, dan angkutan sekolah. Angkutan merupakan salah satu sarana yang berfungsi untuk melakukan perpindahan barang atau orang satu tempat ke tempat yang lainnya. Sedangkan angkutan sekolah merupakan salah satu sarana transportasi yang telah disediakan oleh pihak sekolah untuk siswa-siswi sekolah demi keselamatan dan kelancaran proses belajar mengajar selama jam sekolah (Kariono, 2011). Angkutan sekolah dapat menjadi salah satu pilihan yang dapat mempermudah siswa yang tidak mempunyai kendaraan. Dengan tersedianya bus angkutan sekolah maka dapat menambah tingkat kemacetan di jalan raya. Data yang didapatkan dari Korlantas Polri di tahun 2010-2015 mengemukakan bahwa terdapat 157 ribu anak yang dibawah usia mengalami kecelakaan (Beraunews, 2017).

Salah satu sekolah yang menyediakan pelayanan transportasi angkutan sekolah adalah Sekolah MI Salafiyah Kasim yang berada di kota blitar. Pihak sekolah menyediakan fasilitas tersebut untuk membantu siswa yang jarak rumahnya cukup jauh dari sekolah atau kondisi orang tua yang memiliki kesibukan yang padat. Pihak sekolah menyediakan bus angkutan sekolah ini hanya pada jam pulang sekolah siswa saja. Sehingga dengan adanya fasilitas ini, para siswa dapat dengan nyaman pulang dari sekolah selain itu orang tua para siswa juga merasa lebih tenang karena fasilitas aman yang disediakan oleh pihak

sekolah. Bus angkutan sekolah yang disediakan adalah untuk memfasilitasi pengantaran siswa pada saat jam pulang sekolah. Pengantaran terbagi menjadi dua kloter pemberangkatan yaitu pada pukul 11.30 dan pukul 13.00. Titik awal pengantaran siswa diberangkatkan dari sekolah lalu mengantar siswa ke alamat rumah masing-masing siswa sesuai dengan alamat siswa. Titik akhir bus angkutan sekolah ini harus kembali ke sekolah. Pengambilan data jarak siswa diambil menggunakan maps. Titik awal pengambilan jarak diawali dari sekolah kemudian mengantar siswa pertama, dari alamat siswa pertama akan mengantar ke alamat siswa kedua begitu seterusnya sampai ke alamat siswa terakhir. Jika pengantaran siswa sudah selesai maka bus angkutan sekolah akan kembali ke sekolah.

Pada saat proses pengantaran siswa setiap harinya, siswa yang menaiki bus angkutan sekolah tidak selalu sama atau tetap. Karena itu supir tidak dapat memastikan rute mana yang harus dilewati sehingga tidak dapat tepat waktu kembali ke sekolah untuk kloter selanjutnya. Hal ini menyebabkan siswa pada kloter kedua sering menunggu lama di sekolah karena supir yang terlambat kembali ke sekolah. Dikarenakan supir yang menentukan sendiri rute pengantaran dengan didasari pengalaman pribadi yang mana belum tentu menjadi solusi terbaik sehingga dapat menyebabkan pengeluaran biaya yang mungkin belum diantisipasi oleh pihak sekolah. Sejauh ini pihak sekolah hanya melakukan pendataan siswa yang menaiki angkutan sekolah sesaat sebelum pengantaran siswa sehingga tidak ada cukup waktu untuk menentukan rute pada saat itu juga.

Solusi yang dapat diterapkan pada permasalahan diatas adalah mengoptimasi jarak dengan asumsi semakin pendek jarak yang ditempuh maka dapat meminimalkan waktu pengantaran dan biaya operasionalnya. Salah satu optimasi jarak dapat dilakukan dengan metode algoritme genetika. Optimasi yang bisa dilakukan adalah berdasarkan jarak masing-masing pengantaran siswa. Optimasi dapat dilakukan dengan mengaplikasikan *Travelling Salesman Problem* (TSP). TSP merupakan metode untuk mengkalkulasi jarak dari posisi asal ke tujuan akhir dan kembali ke posisi awal lagi. TSP sebelumnya telah diimplementasikan oleh Dwi Aries Suprayogi (2014) dan Joni dan Nurcahyawati (2012). Optimasi yang dilakukan oleh Dwi Aries Suprayogi (2014) yaitu optimasi antar jemput *laundry* menggunakan nilai *cr* 0,6, *mr* 0,4 dan generasi optimal sebesar 2000. Hasil yang didapatkan dari penelitian ini adalah dapat memberikan solusi untuk melayani semua pelanggan dengan *real time window* masing-masing. Optimasi yang dilakukan oleh Joni dan Nurcahyawati (2012) yaitu pada waktu dan biaya distribusi barang yang menggunakan lima kombinasi kota sebagai tujuan distribusi, dengan nilai *crossover (cr)* dan *mutation (mr)* kurang dari lima puluh, sehingga pada proses inisialisasi menghasilkan nilai *fitness* dari kromosom dengan nilai 931, dan pada proses algoritme genetika menghasilkan nilai *fitness* dengan nilai 762.

Maka berdasarkan penelitian diatas, Joni dan Nurcahyawati (2012) menilai bahwa metode algoritme genetika dapat menghasilkan keluaran yang optimal

dalam pengaplikasian TSP. Berdasarkan hal diatas maka dalam penelitian ini menggunakan TSP dengan Algoritme Genetika sehingga diharapkan dapat menentukan rute terpendek pengantaran siswa menggunakan bus angkutan sekolah.

## 1.2 Rumusan Masalah

Berdasarkan pemaparan latar belakang, maka dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana hasil pengujian ukuran populasi pada optimasi rute pengantaran siswa menggunakan *Algoritme Genetika*?
2. Bagaimana hasil pengujian kombinasi nilai Cr dan Mr pada optimasi rute pengantaran siswa menggunakan *Algoritme Genetika*?
3. Bagaimana hasil pengujian konvergensi pada optimasi rute pengantaran siswa menggunakan *Algoritme Genetika*?
4. Bagaimana tingkat kualitas hasil optimasi *Travelling Salesman Problem* pada kasus angkutan sekolah dengan menggunakan Algoritme Genetika?

## 1.3 Tujuan

Tujuan dilakukan penelitian ini adalah sebagai berikut:

1. Mengetahui hasil optimasi rute pengantaran siswa pada pengujian ukuran populasi.
2. Mengetahui hasil optimasi rute pengantaran siswa pada pengujian kombinasi nilai cr dan mr.
3. Mengetahui hasil optimasi rute pengantaran siswa pada pengujian konvergensi.
4. Mengetahui tingkat kualitas hasil nilai *fitness* dari optimasi *Travelling Salesman Problem* dengan algoritme genetika pada kasus angkutan sekolah.

## 1.4 Manfaat

Manfaat penelitian yang bisa didapat antara lain sebagai berikut:

1. Sekolah dapat mengetahui rute terpendek dalam mengantar pulang siswa.
2. Siswa mendapatkan kenyamanan saat menggunakan fasilitas angkutan sekolah.
3. Dapat mengoptimalkan kinerja supir dalam mengantar.

## 1.5 Batasan Masalah

Agar penelitian ini dapat dilakukan secara optimal, maka diperlukan adanya batasan masalah. Diantaranya adalah sebagai berikut:

1. Data yang digunakan dalam penelitian adalah data siswa yang menggunakan angkutan sekolah yang didapat dari sekolah MI Salafiyah Blitar.

2. Angkutan sekolah yang diteliti dalam penelitian adalah angkutan pribadi dari pihak sekolah itu sendiri.
3. Lokasi pengantaran siswa diasumsikan berada di persimpangan jalan.
4. Penelitian berfokus pada optimasi jarak dikarenakan sudah mewakili biaya bahan bakar dan waktu pengantaran dengan asumsi semakin pendek jarak tempuh maka dapat menekan biaya bahan bakar dan waktu pengantaran.

## 1.6 Sistematika Pembahasan

Sistematika Penulisan yang akan disusun dalam penelitian ini adalah sebagai berikut:

### BAB I PENDAHULUAN

Bab ini akan menguraikan latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah dan sistematika pembahasan yang dilakukan dari penelitian ini.

### BAB II LANDASAN KEPUSTAKAAN

Bab ini akan menguraikan landasan teori dan kajian pustaka dari penelitian-penelitian sebelumnya dan literature lainnya untuk mendukung penelitian ini. Kajian pustaka digunakan sebagai acuan untuk melakukan perbandingan antara penelitian sebelumnya dengan yang dilakukan oleh peneliti. .

### BAB III METODOLOGI

Bab ini akan menguraikan tahapan yang dilakukan dalam penelitian “Optimasi *Travelling Salesman Problem* Pada Angkutan Sekolah Menggunakan Algoritme Genetika”. Tahapan-tahapan yang dilakukan antara lain studi pustaka, pengumpulan data, perancangan, implementasi, pengujian dan kesimpulan dan saran.

### BAB IV PERANCANGAN

Bab ini akan menguraikan proses perancangan sistem dan cara kerja sistem “Optimasi *Travelling Salesman Problem* Pada Angkutan Sekolah Menggunakan Algoritme Genetika”.

### BAB V IMPLEMENTASI

Bab ini akan menguraikan implementasi dari metode yang digunakan, implementasi dari program dan implementasi antarmuka sistem yang akan dibangun dalam penelitian “Optimasi *Travelling Salesman Problem* Pada Angkutan Sekolah Menggunakan Algoritme Genetika”.

### BAB VI PENGUJIAN DAN ANALISIS

Bab ini akan menguraikan proses dan hasil pengujian terhadap metode yang dilakukan dalam penelitian ini. Pengujian yang dilakukan dalam penelitian ini yaitu pengujian akurasi.

### BAB VII PENUTUP



Bab ini akan menguraikan kesimpulan dan saran dari penelitian “Optimasi *Travelling Salesman* Pada Angkutan Sekolah Menggunakan *Algoritme Genetika* (Studi Kasus : MI Salafiyah Kasim Blitar).



## BAB 2 LANDASAN KEPUSTAKAAN

Bab ini akan menguraikan kajian teori dan landasan teori yang akan mendukung penelitian penulis. Kajian pustaka merupakan pembahasan mengenai penelitian-penelitian sebelumnya dengan topik yang terkait. Pada dasar teori menguraikan teori-teori yang berkaitan dengan topik penelitian.

### 2.1 Kajian Pustaka

Kajian pustaka yang digunakan pada penelitian ini merupakan penelitian-penelitian yang membahas topik TSP sehingga dapat digunakan sebagai acuan dalam penelitian ini. Acuan utama yang akan digunakan adalah penelitian dari Khoirun Nisaa pada tahun 2013. Pada penelitian ini penulis bertujuan untuk melakukan optimasi rute angkutan kota malang sehingga meminimalisir jarak tempuh serta penumpukan yang terjadi antar trayek. Yang mana dari *fitness* yang terbaik pada pengujian dengan  $cr = 0,9$ ,  $mr = 0,3$ , jumlah generasi 30 dan populasi 100 sehingga hasil angkutan yang memenuhi jarak tempuh yang ditentukan yaitu sebanyak 19 dari 25 trayek. Kajian pustaka pendukung lainnya yang digunakan antara lain dapat dilihat pada Tabel 2.1.

**Tabel 2.1 Kajian Pustaka**

No	Pustaka	Objek dan Metode	Output
1	Agung Mustika Rizki, Gusti Eka Yuliastuti, Wayan Firdaus Mahmudy, 2017	Distribusi Produk Pada Home Industri Tekstil menggunakan Algoritme Genetika	Pada penelitian ini jumlah populasi yang didapatkan 120, jumlah generasi yang optimal 800, nilai kombinasi $cr$ dan $mr$ sebesar 0,4;0,6 dan nilai <i>fitness</i> sebesar 2,9964. Hasil dari penelitian ini adalah Hasil dari penelitian ini adalah mendapatkan solusi yang optimal sehingga dapat mengefisiensi produk dan dapat mengurangi biaya produksi.
2	Ryan Mahaputra Khrisnanda, Budi Darma Setiawan, Marji, 2017	Penentuan Rute Terpendek Pengambilan Sampah menggunakan M-TSP dan Algoritme Genetika	Pada penelitian ini jumlah populasi yang didapatkan 100, jumlah kendaraan pengangkut sampah yang digunakan sebanyak 4, generasi sebesar 900, nilai kombinasi $cr$ dan $mr$ sebesar 0,3;0,7, dan nilai <i>fitness</i> sebesar 0,569. Hasil dari penelitian ini adalah untuk menemukan rute optimal truk pengangkut sampah yang berangkat dari kantor DKP dengan titik akhir TPA yang melalui titik-titik pengambilan sampah.

3	Cahya Caqiqi, Agus Wahyu Widodo, Yuita Arum Sari, 2017	Wisata Malang Raya menggunakan <i>Evolution Strategies</i> dan <i>Travelling Salesman Problem With Time Windows</i>	Pada penelitian ini jumlah populasi yang didapatkan 100, jumlah generasi 15, nilai <i>fitness</i> sebesar 0,0041. <i>rute</i> yang digunakan ada 15 tetapi ada 2 <i>rute</i> yang tidak dapat teroptimasi. Hasil dari penelitian ini membuktikan bahwa parameter <i>Evolution Strategis</i> dapat mengoptimasi pemilihan rute dengan baik dengan mendapatkan nilai optimasi sebesar 5,57%.
4	Dwi Aries Suprayogi, Wayan F, Mahmudy, 2014	Antar jemput laundry menggunakan Algoritme Genetika dan <i>Travelling Salesman Problem with Time Window</i>	Pada penelitian ini nilai kombinasi <i>crossover</i> dan <i>mutation</i> sebesar 0,4;0,6, generasi yang paling optimal 2000. Hasil yang didapatkan dari penelitian ini adalah dapat memberikan solusi untuk melayani semua pelanggan dengan <i>real time window</i> masing-masing.
5	I Dewa Made Adi Baskara Joni, Vivine Nurcahyawati, 2012.	Jarak Terpendek Pada Jalur Distribusi Barang Di Pulau Jawa Menggunakan Algoritme Genetika	Pada penelitian ini menggunakan 5 kombinasi kota sebagai tujuan distribusi, nilai kombinasi <i>crossover</i> dan <i>mutation</i> lebih kecil dari 50, nilai <i>fitness</i> pada inisialisasi dari kromosom sebesar 931, hasil nilai <i>fitness</i> dari algoritme genetika sebesar 762. Hasil dari penelitian ini adalah dapat mengomptimalkan waktu dan biaya perjalanan
6	Danuri dan Widodo Prijodiprodjo pada tahun 2012	Objek Wisata Daerah Istimewa Yogyakarta menggunakan <i>Bee Colony Optimization Algorithm</i> .	Pada penelitian ini jumlah lebah untuk dilepas antara 10-100 lebah, nilai $\beta$ yang optimal 2, nilai $\lambda$ yang optimal 0,5, dan nilai K yang optimal 1. Hasil dari penelitian ini membuktikan bahwa Algoritme <i>Bee Colony</i> dapat menemukan rute terpendek berdasarkan jumlah lebah yang dilepas.
7	Ryouei Takahashi, 2009	<i>A Hybrid Method of Genetic Algorithms and Ant Colony Optimization to Solve the Traveling</i>	Penelitian ini menjelaskan bahwa <i>Extand Change Crossover Operator (EXCO)</i> yang efisien akan mendapatkan solusi optimal dari permasalahan TSP. Hasil kesimpulan dari penelitian ini adalah bahwa

		<i>Salesman Problem</i>	EXCO dengan menggunakan ACO menghasilkan solusi yang optimum local yang bervariasi, lalu hasil solusi dikirimkan ke <i>Edge Assembly Crossover</i> (EAX) untuk mendapatkan hasil yang optimum yang global sehingga EXCO dapat dilakukan berulang kali untuk konvergensi populasi.
8	H.Afaq, S.Saini, 2012.	<i>A Novel Approach to Solve Graph based Travelling Salesman Problem using Particle Swarm Optimization.</i>	Dalam penelitian ini menyajikan teknik <i>Discrete Optimization</i> (DPSO) pada permasalahan <i>Travelling Salesman Problem</i> (TSP). Teknik yang digunakan dalam penelitian ini adalah mencari jalur tepi yang hilang di dalam grafik sehingga dari <i>tour</i> yang hilang diantara dua node maka akan menghasilkan solusi biaya yang tinggi.
9	Xingfu Wang, Pengcheng Li, Lin Wang, Lei Wang, 2017.	<i>A Novel Genetic Algorithm Based on Circles for Larger-Scale Traveling Salesman Problem.</i>	Dalam penelitian ini menjelaskan bahwa metode algoritme genetika dapat menghasilkan efek yang buruk dalam permasalahan TSP sehingga mengusulkan dua pendekatan yaitu <i>Matrix-Circle</i> (MC) dan <i>Self-Refresh Operation</i> (SRO). Hasil dari pendekatan dua metode tersebut maka menghasilkan kecepatan konvergensi yang lebih tinggi dan nilai yang lebih baik dari TSP skala luar dengan iterasi terbatas.
10	Mahesh Chandra Bhatt, 2017	<i>Travelling Distance Prediction Based Handoff Optimization in Wireless Networks.</i>	Penelitian ini menjelaskan tentang metode untuk mengoptimasi handoff pada <i>wireless network</i> dengan menggunakan <i>travelling distance prediction</i> dan perhitungan threshold. Penelitian ini melakukan perbandingan <i>travelling distance prediction</i> dan metode RSS sehingga dapat ditarik kesimpulan bahwa metode <i>travelling distance prediction</i> lebih efektif dalam mengurangi angka atau jumlah handoff yang tidak diperlukan dan kegagalan handoff

			daripada metode RSS.
11	Penulis	Pemilihan <i>route</i> terpendek pada angkutan sekolah menggunakan <i>Travelling Salesman Problem</i> dan Algoritma Genetika	Hasil yang diperoleh dalam penelitian ini adalah menentukan <i>route</i> mana yang harus dilewati bus angkutan sekolah.

Berdasarkan hasil penelitian-penelitian di Tabel 2.1 dapat dilihat bahwa TSP merupakan salah satu metode yang banyak digunakan dalam menentukan rute. Suprayogi dan Mahmudy (2014) menjelaskan bahwa TSP adalah metode pencarian rute tercepat untuk mencapai tujuan sesuai dengan waktu yang ditentukan dan kembali ke posisi awal. Salah satu metode optimasi yang banyak digunakan dalam implementasi TSP adalah algoritme genetika. Algoritme genetika merupakan salah satu metode dari algoritma evolusi yang dapat menyelesaikan permasalahan optimasi kombinatorial (Dwivedi et.al, 2012). Maka dengan menggunakan TSP dan algoritme genetika dapat menemukan solusi terbaik untuk optimasi rute angkutan sekolah MI Salafiyah Kasim.

## 2.2 MI Salafiyah Kasim

MI Salafiyah Kasim adalah salah satu sekolah pendidikan dasar yang berdiri pada tanggal 30 Januari 1981. Ini merupakan sekolah Lembaga Pendidikan Madrasah Nahdlatul Ulama (LPM NU) yang berada di kota Blitar tempatnya di jalan KH. Dimiyati Kasim Ploso, kecamatan Selopuro. Berikut bangunan gedung sekolah MI Salafiyah Kasim dapat ditunjukkan pada Gambar 2.1.



Gambar 2.1 (a) Tampilan Depan Gedung Sekolah, (b) Tampilan Samping Gedung Sekolah



Sekolah dasar ini merupakan salah satu sekolah yang menyediakan fasilitas angkutan sekolah. Sistem yang diterapkan oleh pihak sekolah hanya sistem pengantaran siswa pada saat jam pulang sekolah saja, itu sudah menjadi kesepakatan antara pihak sekolah dengan orang tua siswa. Angkutan sekolah yang disediakan oleh pihak sekolah hanya ada satu dan sudah mulai beroperasi selama 3 tahun. Selain menyediakan fasilitas angkutan sekolah, sekolah dasar ini juga mempunyai program unggulan dan program pembelajaran seperti Tahfidzh Al Qur'an, Pembelajaran Al Qur'an Metode "TILAWATI", Penambahan jam belajar setelah jam pulang sekolah dan bimbingan belajar. Berikut Bus Angkutan Sekolah yang disediakan oleh pihak sekolah dapat ditunjukkan pada Gambar 2.2.



Gambar 2.2 (a) Tampilan Depan Bus Sekolah, (b) Tampilan Dalam Bus Sekolah

### 2.3 Travelling Salesman Problem (TSP)

TSP merupakan salah satu bagian dari permasalahan optimasi kombinatorial untuk menentukan jarak terpendek yang akan dilewati dalam rute perjalanan dimana masing-masing kota hanya dapat dilewati satu kali saja kemudian akan kembali ke kota awal (Erdiwansyah dan Gani, 2016). Ada beberapa macam karakteristik pada TSP yaitu semua kota harus dilewati tanpa terkecuali, tidak dapat kembali ke kota awal jika masih ada kota yang belum dilewati, dan yang terakhir setiap kota hanya bisa dilewati satu kali saja. Tujuan TSP dari permasalahan diatas adalah untuk dapat membangun rute yang optimal dengan meminimalkan total jarak yang ditempuh dalam rute perjalanan sehingga dapat menghemat biaya.

Pada umumnya, TSP dapat dibagi menjadi 2 bagian yaitu TSP Simetris (STSP) dan TSP asimetris (ATSP). Bentuk TSP yang simetris adalah jarak yang dilalui hanya ada satu jarak sehingga jarak antara kota A dan kota B sama dengan jarak antara kota B dan kota A, sedangkan bentuk TSP asimetris merupakan kebalikan dari TSP simetris yang mempunyai dua jarak sehingga jarak antara kota A ke kota B tidak sama dengan jarak dari kota B dan kota A.



Rumus Permutasi TSP yang bisa digunakan adalah sebagai berikut:

$$p_n^k = \frac{n!}{(n-k)!} \quad (2.1)$$

Keterangan:

$n$  = jumlah semua node.

$k$  = jumlah node yang dilalui.

## 2.4 Optimasi

Pada teknik optimasi, data yang dapat diolah adalah data yang memiliki ukuran yang kecil. Karena jika data yang diolah memiliki ukuran yang besar maka penyelesaiannya akan semakin lebih kompleks dan membutuhkan waktu yang lama untuk diselesaikan (Dorigo dan Stutzle, 2004). Optimasi merupakan proses yang dilakukan dengan cara memilih solusi yang terbaik dari banyaknya solusi-solusi yang ada dengan cara memenuhi sejumlah batasan (Mahmudy, 2013).

## 2.5 Algoritme Genetika

Dalam ilmu *Algorithm Evolution*, algoritme genetika merupakan metode yang cukup banyak digunakan untuk menemukan solusi optimasi (Mahmudy, 2015). Bremermann dan John Halland merupakan salah satu matematikawan yang mengembangkan algoritme genetika tahun 1975 di Universitas Michigan, New York. Algoritme genetika merupakan algoritme yang dapat memecahkan permasalahan pada optimasi kombinatorial (Potvin, 1996). Di dalam algoritme genetika terdapat individu-individu yang disebut dengan *chromosome*. *Chromosome* merupakan kumpulan alternative solusi dari setiap masalah (Hannawati, Thiag & Eleazar, 2002). Tahapan pada algoritma genetika dapat dilihat pada Gambar 2.3.



Gambar 2.3 Tahapan Algoritme Genetika

### 2.5.1 Inisialisasi (Populasi Awal)

Inisialisasi (populasi awal) merupakan proses tahapan awal yang pertama kali dilakukan pada algoritme genetika. Pada proses ini populasi awal beranggotakan individu-individu (*chromosome*) yang dibentuk secara acak. *Chromosome* berisi solusi dari permasalahan-permasalahan yang ingin diselesaikan. Pada penelitian ini proses inisialisasi dilakukan dengan menggunakan teknik kombinatorial. Terdapat beberapa cara untuk membangkitkan inisialisasi yaitu permutasi gen, algoritme *random generator* dan lain-lain.

## 2.5.2 Reproduksi

Reproduksi merupakan proses tahapan kedua yang dilakukan pada alur algoritme genetika. Pada proses reproduksi akan menghasilkan individu-individu baru dari individu yang sudah ada. Individu yang dihasilkan akan mewarisi sifat dari induk. Reproduksi dilakukan dengan tujuan untuk menghasilkan individu baru agar menjaga sifat baik dari induk (Hannawati, Thiang & Eleazar, 2002). Reproduksi dibagi menjadi 2 tahap yaitu:

### A. Crossover

Crossover (tukar silang) merupakan proses reproduksi yang pertama kali dilakukan dalam algoritme genetika. Pada penelitian ini akan menggunakan *crossover one-cut point*. Proses *crossover* tersebut dilakukan dengan cara menentukan induk dan titik potong secara acak, selanjutnya dilakukan pertukaran gen antar induk berdasarkan titik potong yang ditentukan (Mahmudy, 2015). Untuk mengetahui jumlah *crossover* adalah dengan cara melakukan perkalian antara *crossover rate* (*cr*) dengan *popsiz*e sehingga dapat diketahui jumlah individu baru yang dihasilkan. Misal jika *cr* = 0,5 dan *popsiz*e = 3 maka akan ada 1 *offspring* yang dihasilkan. Berikut adalah contoh reproduksi *one cut point crossover*, misalkan *parent* yang dipilih P1 dan P3 maka *offspring* yang didapatkan C1 dapat ditunjukkan pada Gambar 2.4.

P1	2	3	9	20	4	10	19	15	6	14	8	11	17	7	15	18	13
P3	11	8	4	20	14	10	7	9	18	13	17	5	19	2	14	3	15

Cut point di titik ke-7

C1	2	3	9	20	4	10	19	11	8	14	7	18	13	17	5	14	15
----	---	---	---	----	---	----	----	----	---	----	---	----	----	----	---	----	----

Gambar 2.4 Reproduksi Crossover One Cut Point

### B. Mutasi

Mutasi merupakan proses reproduksi kedua setelah proses *crossover*. Dalam penelitian ini, proses mutasi yang dilakukan yaitu dengan menggunakan *exchange mutation*. Proses mutasi ini dilakukan dengan cara menukar dua titik yang akan dimutasi lalu nilainya ditukar. Untuk mengetahui jumlah *mutation* adalah dengan cara melakukan perkalian antara *mutation rate* dengan *popsiz*e sehingga dapat diketahui jumlah *offspring* yang dihasilkan. Contoh *exchange mutation* dapat ditunjukkan pada Gambar 2.5.

P2	10	20	2	13	5	7	11
C2	10	11	2	13	5	7	20

**Gambar 2. 5 Reproduksi *Exchange Mutation***

### 2.5.3 Evaluasi

Evaluasi merupakan proses tahapan ketiga yang dilakukan pada algoritme genetika. Evaluasi dilakukan untuk menentukan individu terbaik dengan cara menggabungkan nilai *fitness* dari tiga proses sebelumnya (inisialisasi, *crossover* dan mutasi). Nilai *fitness* merupakan nilai untuk menentukan individu mana yang terbaik atau terburuk. Jika nilai *fitness* semakin besar maka semakin baik digunakan sebagai solusi di dalam *chromosome*. Menurut Krishnanda (2013), terdapat dua persamaan yaitu, yang pertama adalah persamaan untuk memaksimalkan nilai *fitness* dan yang kedua untuk meminimalkan nilai *fitness*. Rumus nilai *fitness* yang digunakan adalah sebagai berikut:

$$Fitness = \frac{100}{Total \text{ Jarak yang dilewati}} \quad (2.2)$$

### 2.5.4 Seleksi

Tahapan selanjutnya adalah seleksi. Dimana tahapan ini merupakan tahapan terakhir pada algoritme genetika. Pada penelitian ini menggunakan metode *elitism selection*. Metode *elitism selection* merupakan metode untuk melakukan seleksi pada individu terbaik sehingga dapat digunakan pada generasi selanjutnya berdasarkan nilai *fitness* tertinggi. Ketika proses seleksi selesai dilakukan akan menghasilkan populasi baru. Jika nilai *fitness* dari populasi baru semakin tinggi maka akan lebih baik (Hannawati, 2011).

## BAB 3 METODOLOGI

Pada bab ini menguraikan tentang tahapan-tahapan yang dilakukan untuk optimasi *travelling salesman problem* (TSP) pada angkutan sekolah menggunakan metode algoritme genetika. Tahapan-tahapan yang digunakan dalam proses optimasi jarak ini secara umum yakni sebagai berikut Tipe Penelitian, Strategi Penelitian, Partisipan Penelitian, Lokasi Penelitian, Teknik Pengumpulan Data Implementasi Algoritme, dan Teknik Analisis Data.

### 3.1 Tipe Penelitian

Tipe penelitian yang digunakan dalam penelitian ini adalah Analitik. Tipe penelitian Analitik lebih berfokus terhadap fenomena yang sedang diteliti untuk menghasilkan rekomendasi rute dengan jarak terpendek. Pada penelitian ini menggunakan jenis penelitian kualitatif dan kuantitatif dengan menggunakan data siswa dan alamat siswa yang bertujuan untuk menemukan rute terpendek.

### 3.2 Strategi Penelitian

Strategi penelitian yang digunakan dalam penelitian ini yaitu menggunakan metode eksperimen. Metode eksperimen yang digunakan ditujukan untuk mengkaji suatu pengaruh variabel tertentu terhadap variabel lain dalam kondisi yang terkontrol dengan ketat. Selain itu strategi penelitian dengan memilih studi kasus pada tempat tertentu ditujukan untuk menggali permasalahan yang penyelesaiannya dapat diselesaikan oleh metode pembelajaran

### 3.3 Partisipan Penelitian

Partisipan yang terlibat dalam penelitian ini adalah guru sekolah dasar untuk mendapatkan data siswa. Alasan memilih guru sekolah dasar adalah agar informasi yang didapatkan lebih akurat. Informasi didapatkan dengan cara melakukan wawancara langsung kepada pihak guru sekolah MI Salafiyah Kasim.

### 3.4 Lokasi Penelitian

Penelitian ini dilakukan di Sekolah MI Salafiyah Kasim Blitar. Alasan memilih lokasi penelitian dikarenakan sekolah tersebut merupakan salah satu sekolah yang memiliki bus angkutan sekolah.

### 3.5 Teknik Pengumpulan Data

Pada tahap ini teknik pengumpulan data dilakukan dengan tujuan untuk mengumpulkan data yang akan digunakan selama penelitian. Pada penelitian ini, mengambil data dari sekolah MI Salafiyah Kasim yang terletak di Blitar yang akan digunakan pada penelitian ini. Dari data tersebut akan mencari berapa jarak dari lokasi sekolah ke masing-masing rumah siswa pada saat jam pengantaran siswa menggunakan *Travelling Salesman Problem*. Setelah mendapatkan jarak lalu melakukan optimasi jarak menggunakan algoritme genetika untuk mengetahui

rute mana yang harus dilewati bus angkutan sekolah pada jam pengantaran siswa. Penelitian ini menggunakan data kualitatif dan kuantitatif. Dimana data kualitatif pada penelitian ini berupa data siswa dan alamat siswa yang didapatkan dari wawancara. Sedangkan data kuantitatif didapatkan dari pengkonversian data siswa dan alamat siswa menjadi id berupa angka.

### 3.6 Implementasi Algoritme

Implementasi algoritme terdiri dari beberapa proses yaitu proses manualisasi, perancangan sistem, dan implementasi sistem. Proses manualisasi bertujuan untuk mengetahui tahap-tahap perhitungan untuk membuat perancangan sistem. Perancangan sistem merupakan pembuatan alur sistem akan digunakan sebagai panduan pengembangan sistem, sedangkan implementasi merupakan proses pembangunan sistem sehingga didapatkan hasil penelitian berupa optimasi rute pada angkutan sekolah.

### 3.7 Teknik Analisis Data

Pada tahap ini teknik analisis data dilakukan dengan melakukan proses pengujian menggunakan metode algoritme genetika. Hasil perhitungan data yang digunakan didapatkan dari 4 proses pengujian yaitu *crossover*, *mutation*, evaluasi dan seleksi. Dari hasil pengujian tersebut diperoleh hasil terbaik dalam melakukan optimasi jarak.

### 3.8 Jadwal Penelitian

Jadwal penelitian akan dilaksanakan dalam waktu satu semester terhitung dari mulai bulan Februari sampai bulan Juni. Berikut jadwal penelitian akan ditunjukkan pada Tabel 3.1.

**Tabel 3.1 Jadwal Penelitian**

No	Uraian	Februari				Maret				April				Mei				Juni			
		Minggu ke -																			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Studi Kepustakaan																				
2	Pengumpulan Data																				
3	Implementasi Algoritme																				

## BAB 4 PERANCANGAN

Pada bab ini akan menguraikan tentang formulasi permasalahan, siklus penyelesaian permasalahan dengan menggunakan algoritme genetika, proses perhitungan manualisasi, dan perancangan antarmuka pada sistem.

### 4.1 Formulasi Permasalahan

Permasalahan yang diangkat pada penelitian ini adalah optimasi rute terpendek pada angkutan sekolah dengan menggunakan data siswa dan alamat siswa. Jam pengantaran siswa pada angkutan sekolah MI Salafiyah Kasim terbagi menjadi dua bagian antara lain kloter pagi dan kloter siang. Jumlah data siswa yang menaiki bus angkutan sekolah pada kloter pagi sebanyak 20 siswa dan pada kloter siang sebanyak 24 siswa. Data siswa yang menaiki bus angkutan sekolah pada jam pengantaran siswa dapat dilihat pada Tabel 4.1.

**Tabel 4.1 Data Siswa Kloter Pagi**

Shift Pagi				
ID	Nama	panggilan	Kelas	Alamat
1	ANAK 1	1	1	Bukur 01/05 Popoh
2	ANAK 2	2	2	Bukur 02/05 Popoh
3	ANAK 3	3	1	Bukur 02/05 Popoh
4	ANAK 4	4	2	Bukur 02/05 Popoh
5	ANAK 5	5	2	Bukur 02/05 Popoh
6	ANAK 6	6	1	Bukur 02/05 Popoh
7	ANAK 7	7	1	Bukur 03/04 Popoh
8	ANAK 8	8	1	Bukur 03/05 Popoh
9	ANAK 9	9	1	Bukur 05/05 Popoh
10	ANAK 10	10	1	Bukur 05/05 Popoh
...	....	....	...	....
20	ANAK 20	20	1	Ploso 03/03 Selopuro

**Tabel 4.2 Data Siswa Kloter Siang**

Shift Siang				
ID	Nama	Panggilan	Kelas	Alamat
1	ANAK 1	1	4	Bukur 02/05 Popoh
2	ANAK 2	2	5	Bukur 03/05 Popoh
3	ANAK 3	3	6	Darungan 04/05
4	ANAK 4	4	3	Gading
5	ANAK 5	5	5	Gading
6	ANAK 6	6	5	Gading
7	ANAK 7	7	4	Jatikeplek 01/06 Klemunan
8	ANAK 8	8	4	Jatikeplek 02/06 Klemunan
9	ANAK 9	9	4	Jatikeplek 03/06 Klemunan
10	ANAK 10	10	3	Klemunan 03/06 Wlingi
11	ANAK 11	11	6	Jepun 04/06 Tegalrejo

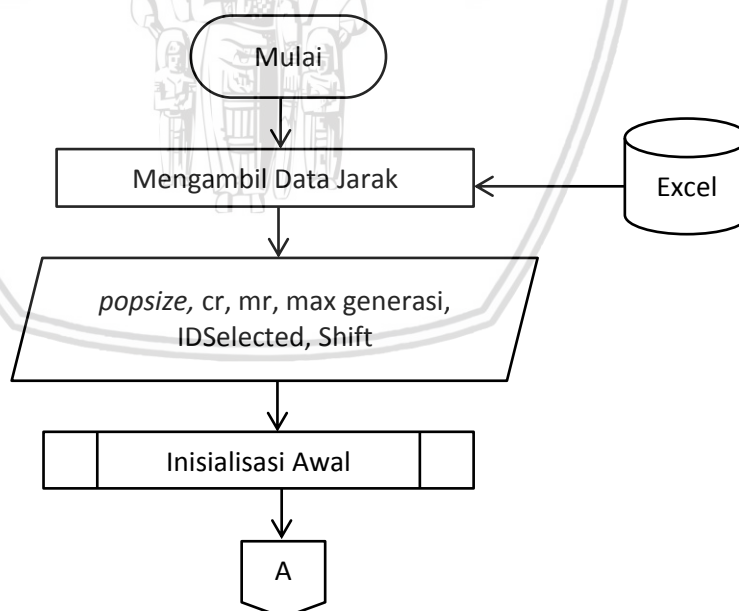


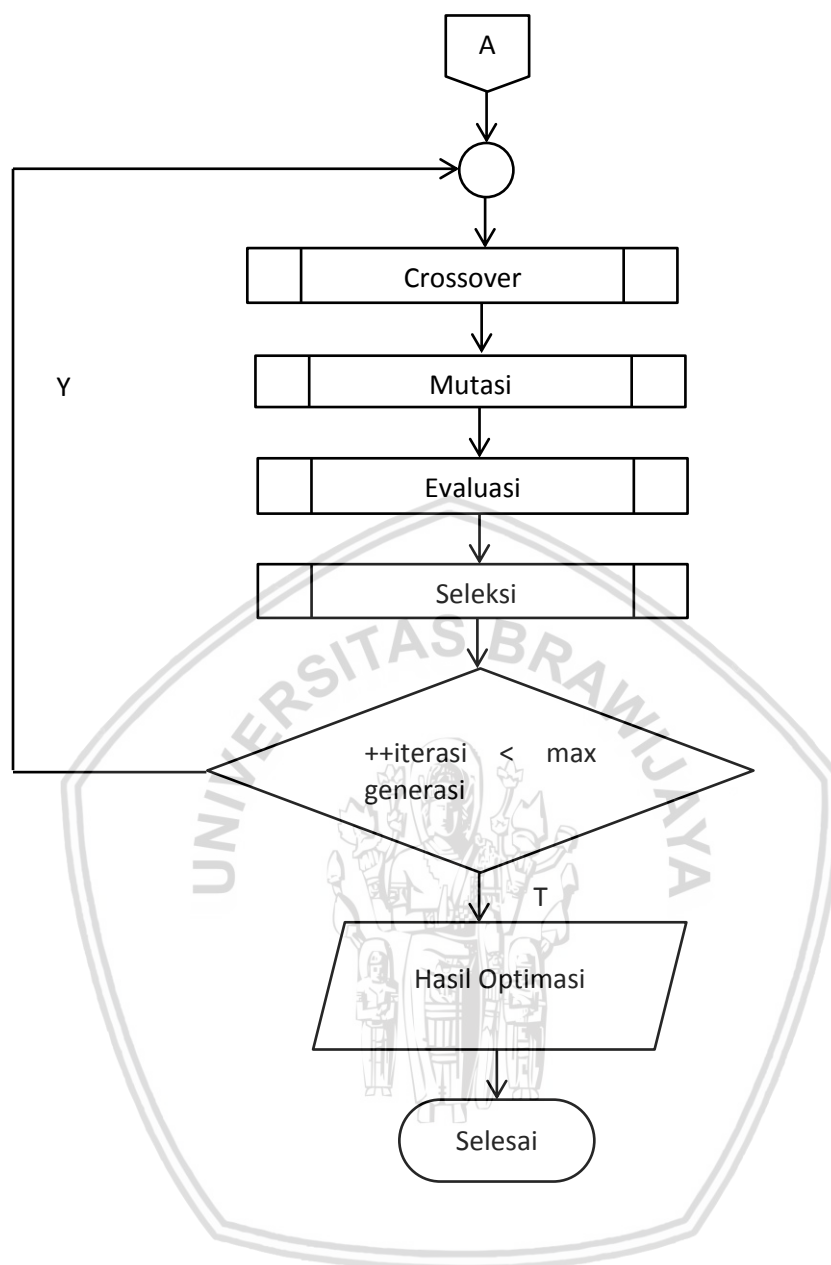
12	ANAK 12	12	3	Jepun 04/06 Tegalrejo
...	....	....	...	....
24	ANAK 24	24	3	Popoh 02/02 Popoh

Pada Tabel 4.1 dan Tabel 4.2 terdapat beberapa siswa dengan alamat rumah yang identik sama untuk RT dan RW nya. Pengambilan matriks jarak siswa berdasarkan titik lokasi rumah masing-masing siswa secara *real*. Meskipun terdapat alamat rumah siswa nya sama namun pada kenyataannya lokasi rumah tersebut berbeda blok atau gang pada desanya.

## 4.2 Proses Perancangan Algoritme Genetika

Untuk melakukan optimasi rute pada permasalahan ini menggunakan TSP dan algoritme genetika. TSP bertujuan untuk mengetahui jumlah jarak pengantaran dari masing-masing siswa. Selanjutnya hasil dari TSP akan digunakan dalam proses algoritme genetika. Pada algoritme genetika terdapat 4 proses yang pertama inisialisasi awal untuk mengidentifikasi solusi dari permasalahan. Yang kedua proses reproduksi terdiri dari 2 proses yaitu *crossover* dan *mutation* yang mana sama-sama bertujuan untuk mendapatkan *offspring*. Selanjutnya pada proses evaluasi bertujuan untuk mendapatkan nilai *fitness* dari masing-masing individu dan terakhir adalah proses seleksi yang bertujuan untuk mendapatkan individu baru berdasarkan nilai *fitness* mulai dari yang tertinggi sampai terendah. Diagram alir dari algoritme genetika dapat dilihat pada Gambar 4.1.



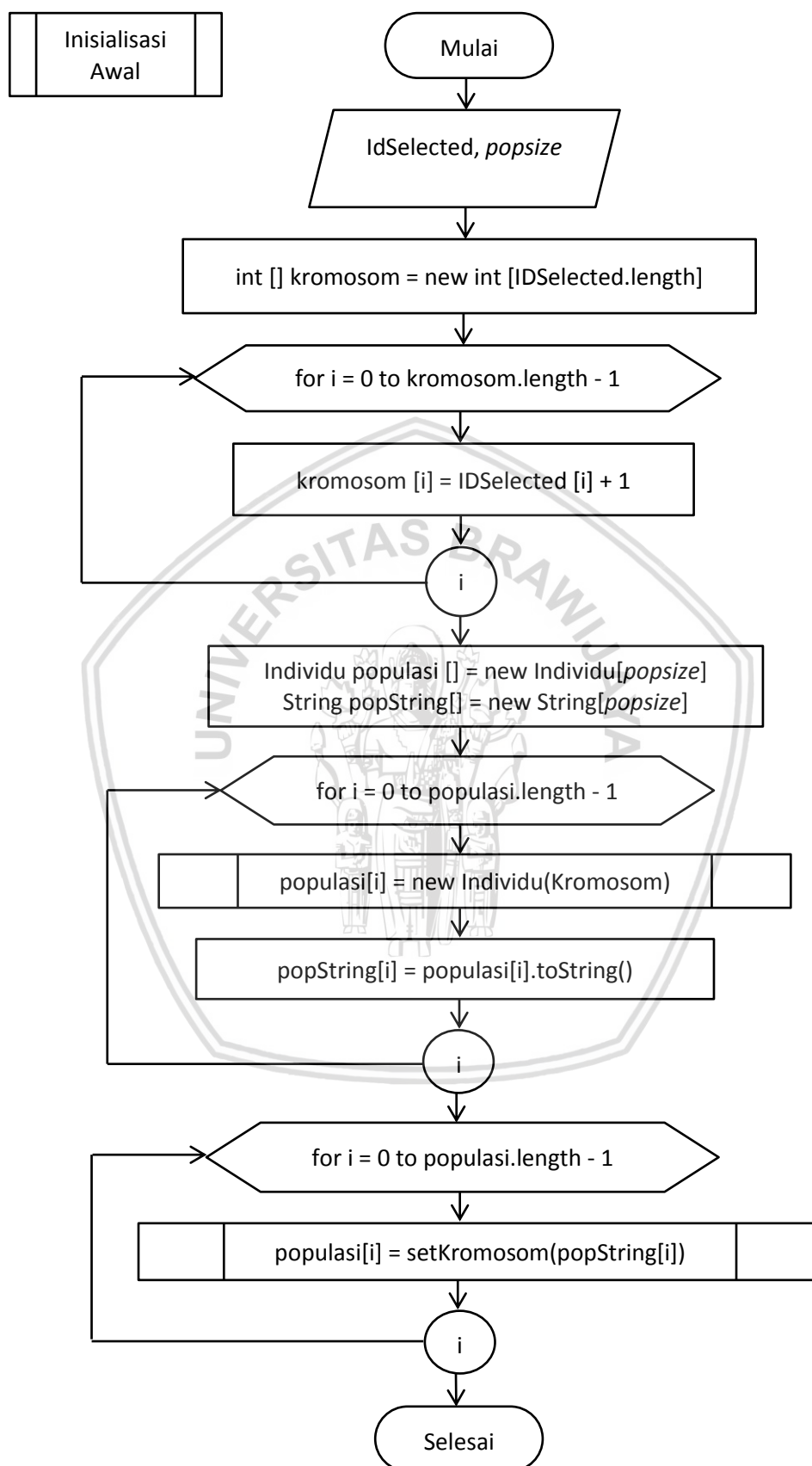


Gambar 4.1 Diagram Alir Algoritme Genetika Pada Rute Angkutan Sekolah

### 4.3 Tahapan-tahapan Algoritme Genetika (GA)

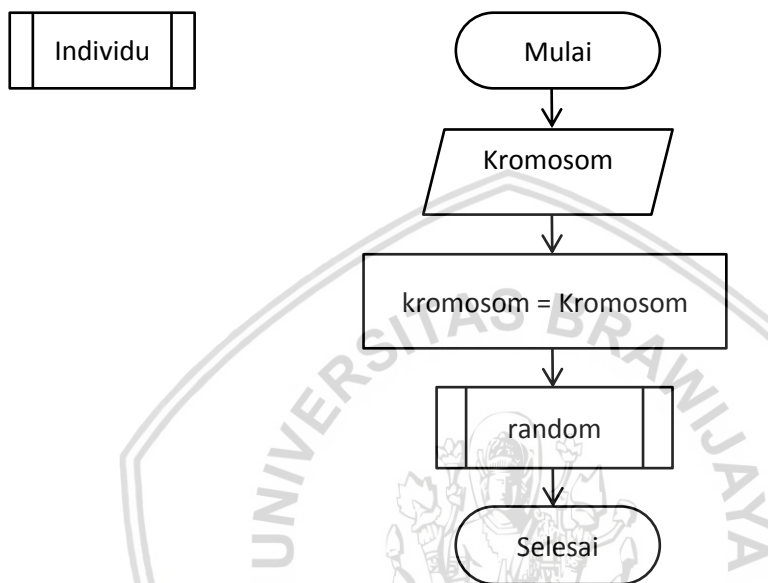
#### 4.3.1 Inisialisasi Populasi Awal

Inisialisasi awal merupakan tahapan awal pada proses algoritme genetika. Proses inisialisasi awal dilakukan dengan cara mengambil *parent* sebanyak jumlah *popsi*, kemudian mengacak nilai individu pada masing-masing *parent* yang sudah didapatkan.



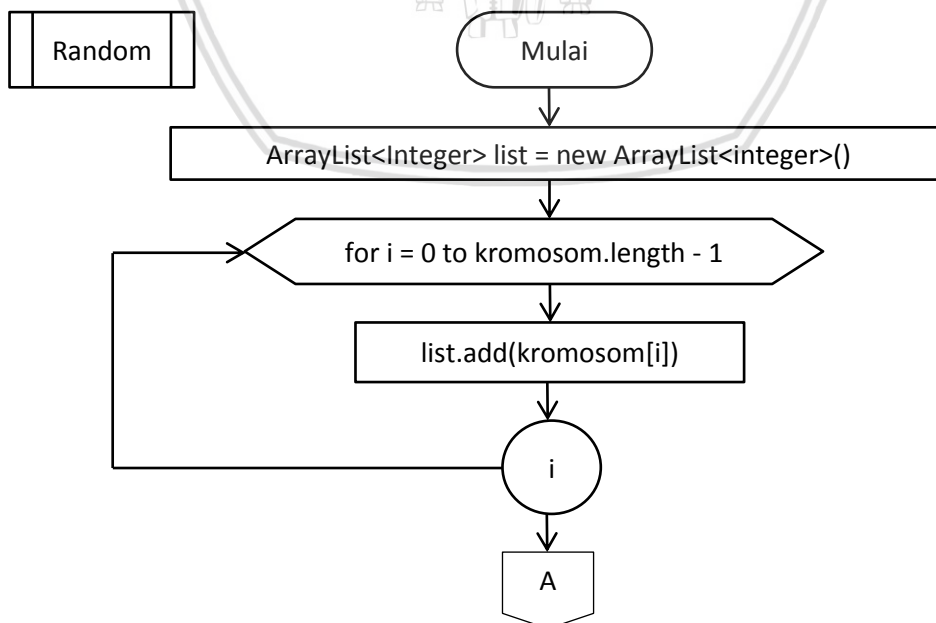
Gambar 4.2 Diagram Alir Inisialisasi Awal

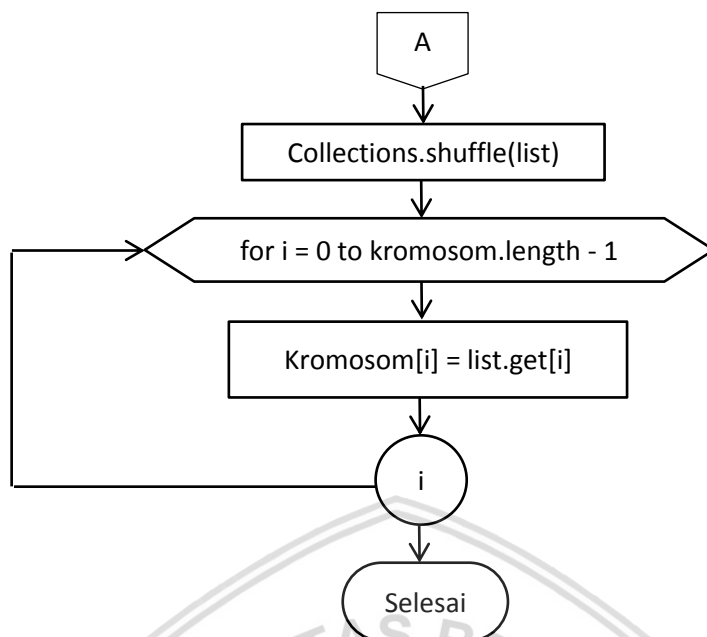
Gambar 4.2 merupakan gambar diagram alir untuk proses inialisasi awal. Pada proses inialisasi awal membutuhkan masukan berupa *IdSelected* dan *popsiz*. *IdSelected* adalah id siswa yang sudah dipilih oleh *user* yang nantinya akan disimpan dalam *array* kromosom. Pembangkitan populasi dilakukan sesuai dengan masukan dari *user* dengan nilai *random*. Nilai kromosom akan disimpan sementara dalam variabel *popstring* sebelum dikembalikan lagi ke dalam populasi.



**Gambar 4.3 Diagram Alir Individu**

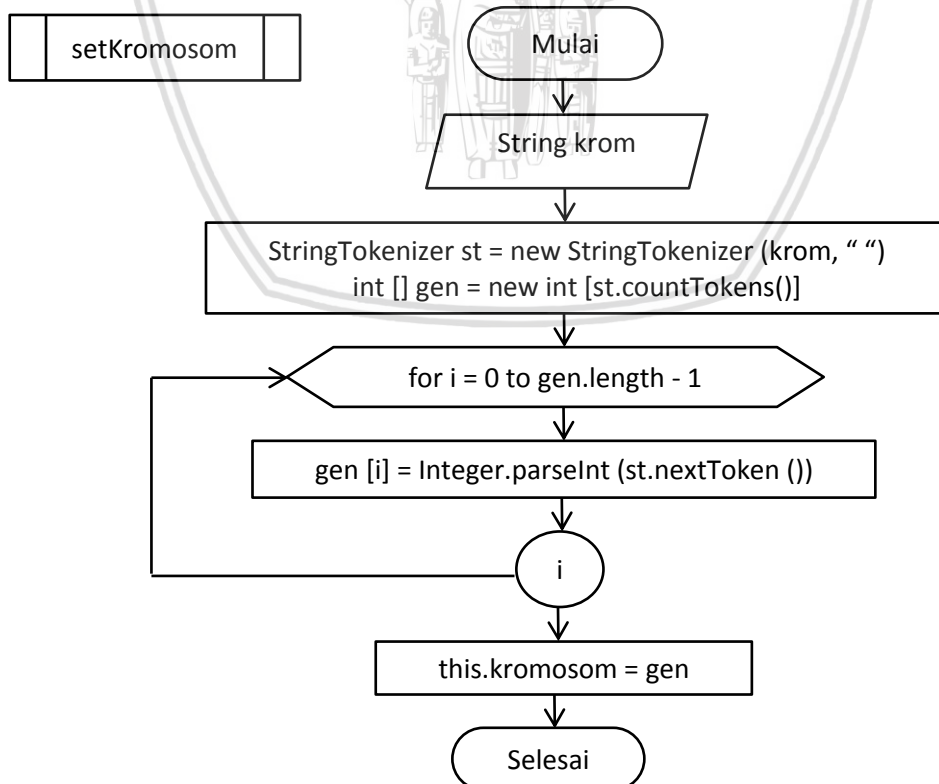
Gambar 4.3 merupakan gambar alir individu. Proses ini untuk membuat individu baru yang dibangkitkan secara acak berdasarkan masukan dari *user*.





**Gambar 4.4 Diagram Alir *Random***

Gambar 4.4 merupakan gambar alir *random*. Proses ini dilakukan untuk mengacak isi dari kromosom dengan cara melakukan perulangan sebanyak panjang kromosom untuk memasukkan nilai kromosom ke dalam *array list*. Kemudian proses pengacakan dengan menggunakan *library collections*. Setelah itu melakukan perulangan lagi untuk mengembalikan nilai acak ke kromosom.

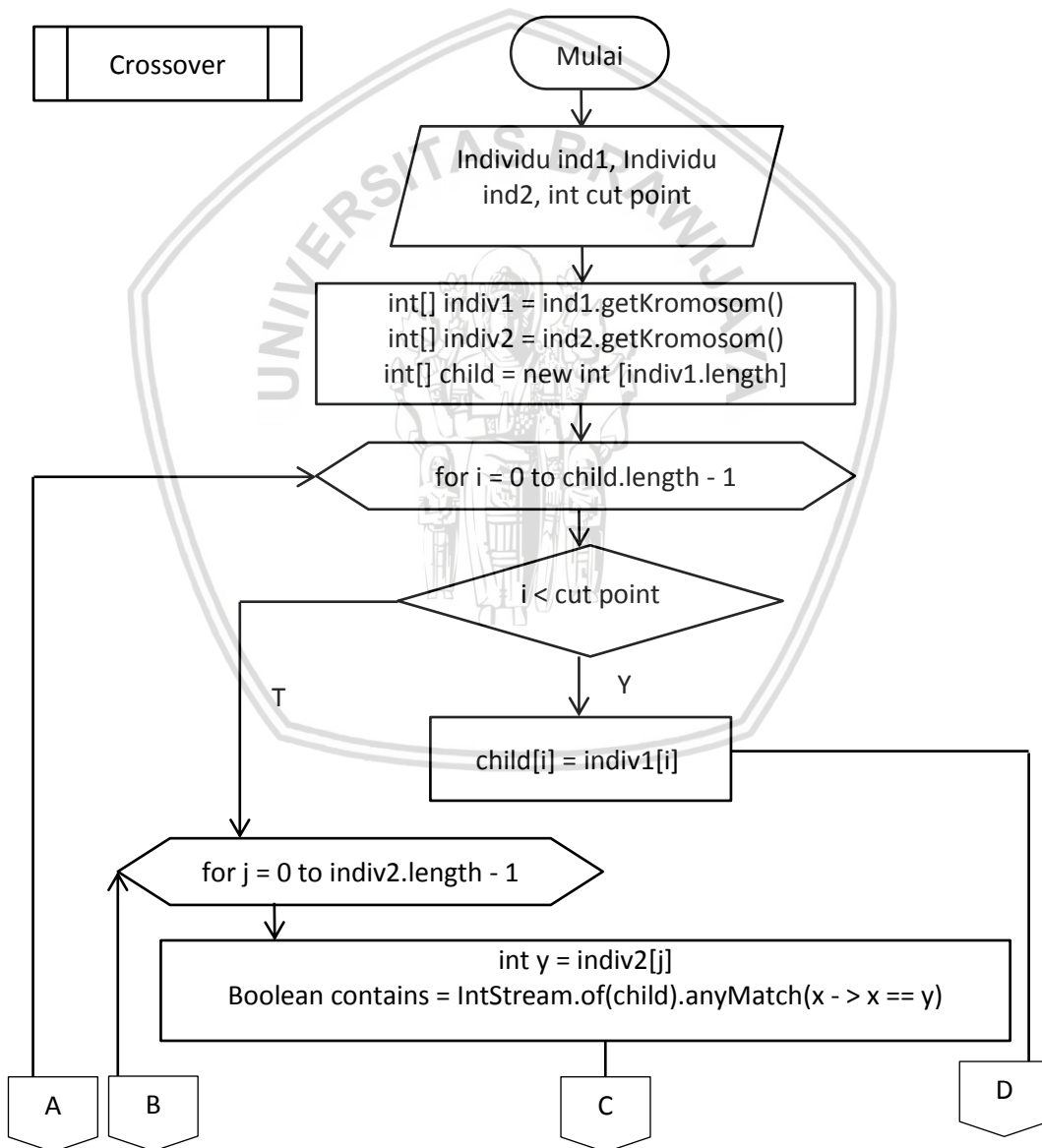


**Gambar 4.5 Diagram Alir setKromosom**

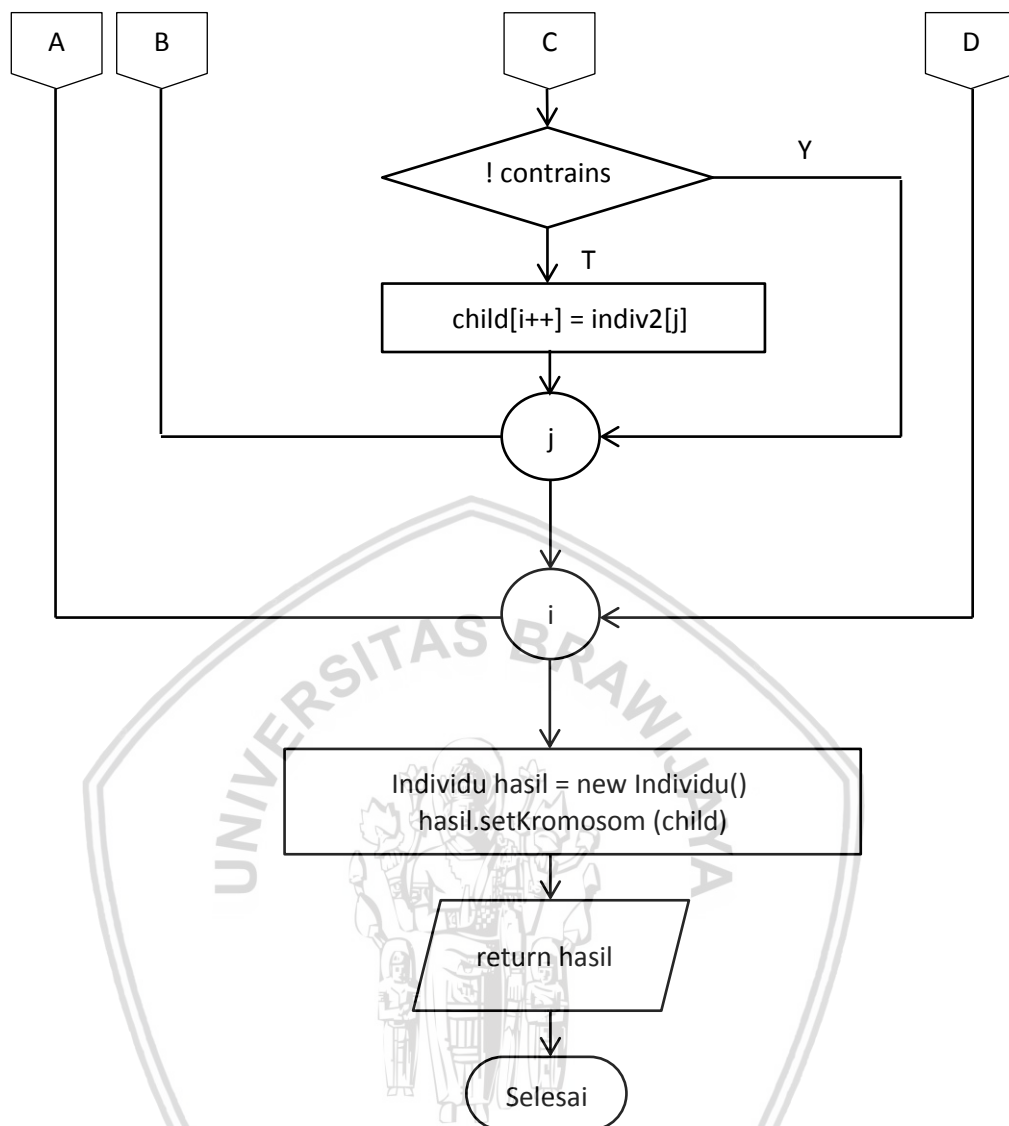
Gambar 4.5 merupakan gambar diagram alir *setKromosom*. Proses ini dilakukan dengan tujuan untuk dapat melakukan *setKromosom* suatu individu dengan masukan berupa *string*.

#### 4.3.2 Crossover

Tahapan kedua adalah proses reproduksi yang terbagi menjadi dua yaitu *crossover* dan mutasi. Proses *crossover* dapat dilihat pada Gambar 4.6. Pada proses ini dilakukan dengan tujuan untuk mendapatkan *offspring*. Proses *crossover* dilakukan dengan cara mengambil secara acak dua parent yang sudah terbentuk pada proses inisialisasi awal. Pembentukan *offspring* menggunakan metode *one cut point crossover*. Diagram alir *crossver* dapat dilihat pada Gambar 4.6.

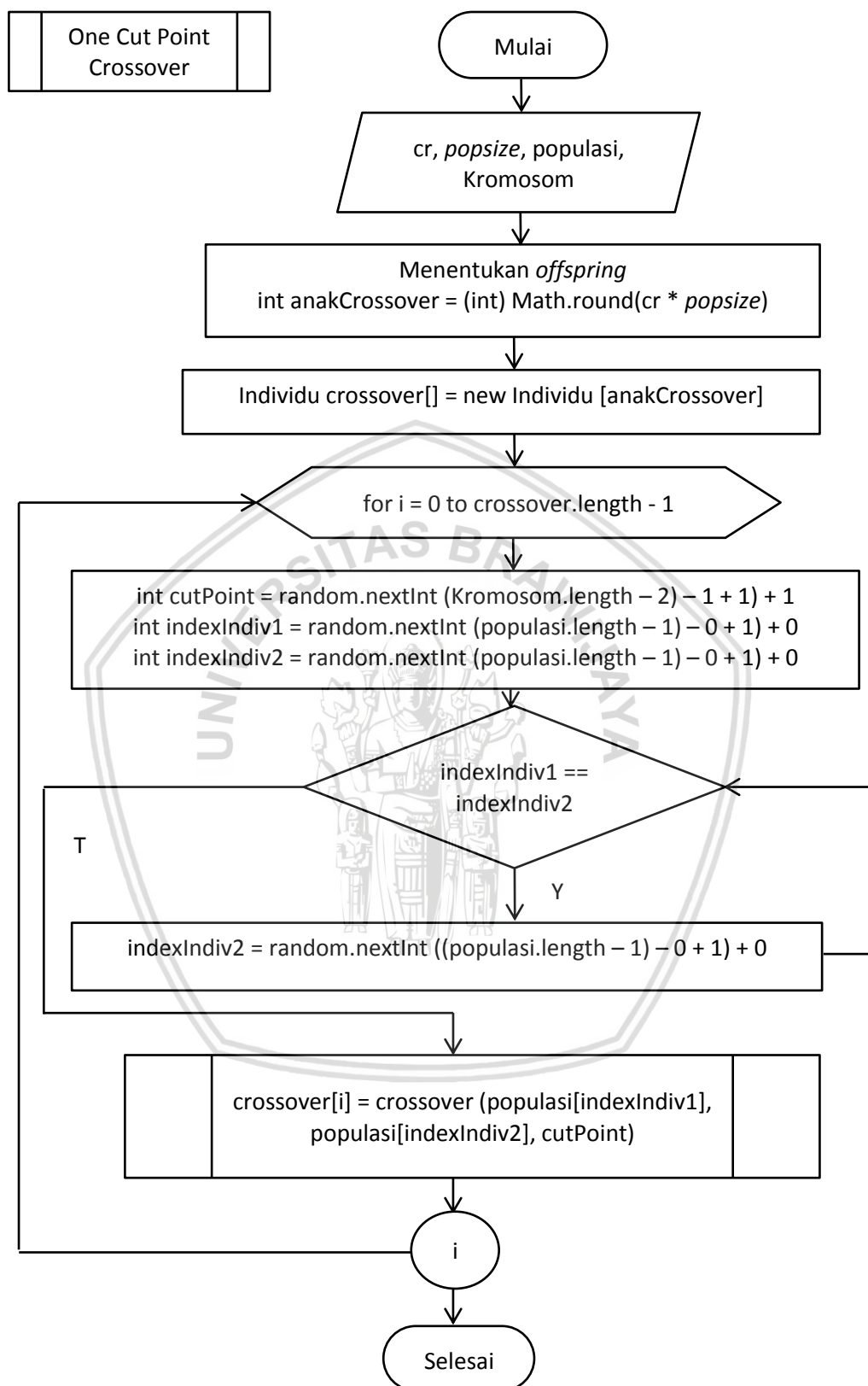






**Gambar 4.6 Diagram Alir Crossover**

Pada Gambar 4.6 proses *crossover* diawali dengan memasukkan individu ind1, individu ind2, dan int *cut point*. Individu ind1 dan individu ind2 merupakan variabel *parent* pertama dan *parent* kedua yang dipilih secara acak kemudian melakukan proses penyimpanan kromosom pada indiv1 (*parent* pertama), indiv2 (*parent* kedua). Selanjutnya menyimpan hasil kromosom dari *parent* pertama dan *parent* kedua dalam proses *child*. Constraint merupakan variabel Boolean untuk menyimpan nilai ya jika kromosom *parent* kedua tidak terdapat pada *child* dan tidak jika terdapat pada *child* dan return hasil yaitu variabel untuk menyimpan hasil *crossover*.

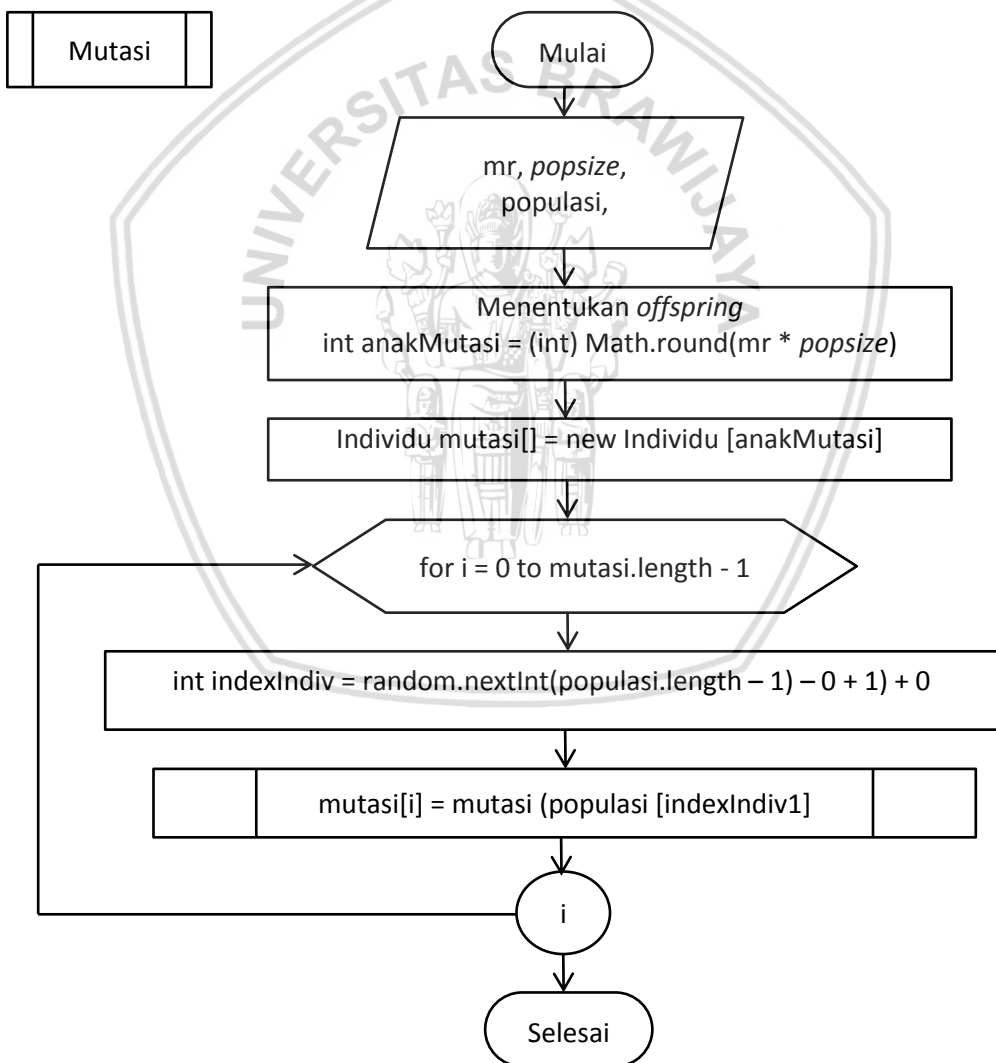


Gambar 4.7 Diagram Alir One Cut Point Crossover

Gambar 4.7 merupakan proses *one cut point crossover* yang diawali dengan menginputkan nilai *cr*, *popsiz*, populasi dan kromosom. Proses ini dilakukan dengan cara menentukan jumlah *offspring* yang akan dibentuk. Selanjutnya akan melakukan perulangan untuk menghitung *crossover* yang dimulai dari 0 sampai panjang *crossover*. Proses untuk mencari *cut point*, *IndexIndiv1* (*parent 1*) dan *indexIndiv2* (*parent 2*) dilakukan secara acak. Kemudian melakukan perulangan lagi untuk merubah *indexIndiv2* jika memiliki nilai yang sama dengan *indexIndiv1*. Setelah itu menentukan nilai *array crossover* dari *method crossover* dengan parameter populasi ke-*indexIndiv1*, populasi ke-*indexIndiv2* dan *cutpoint*.

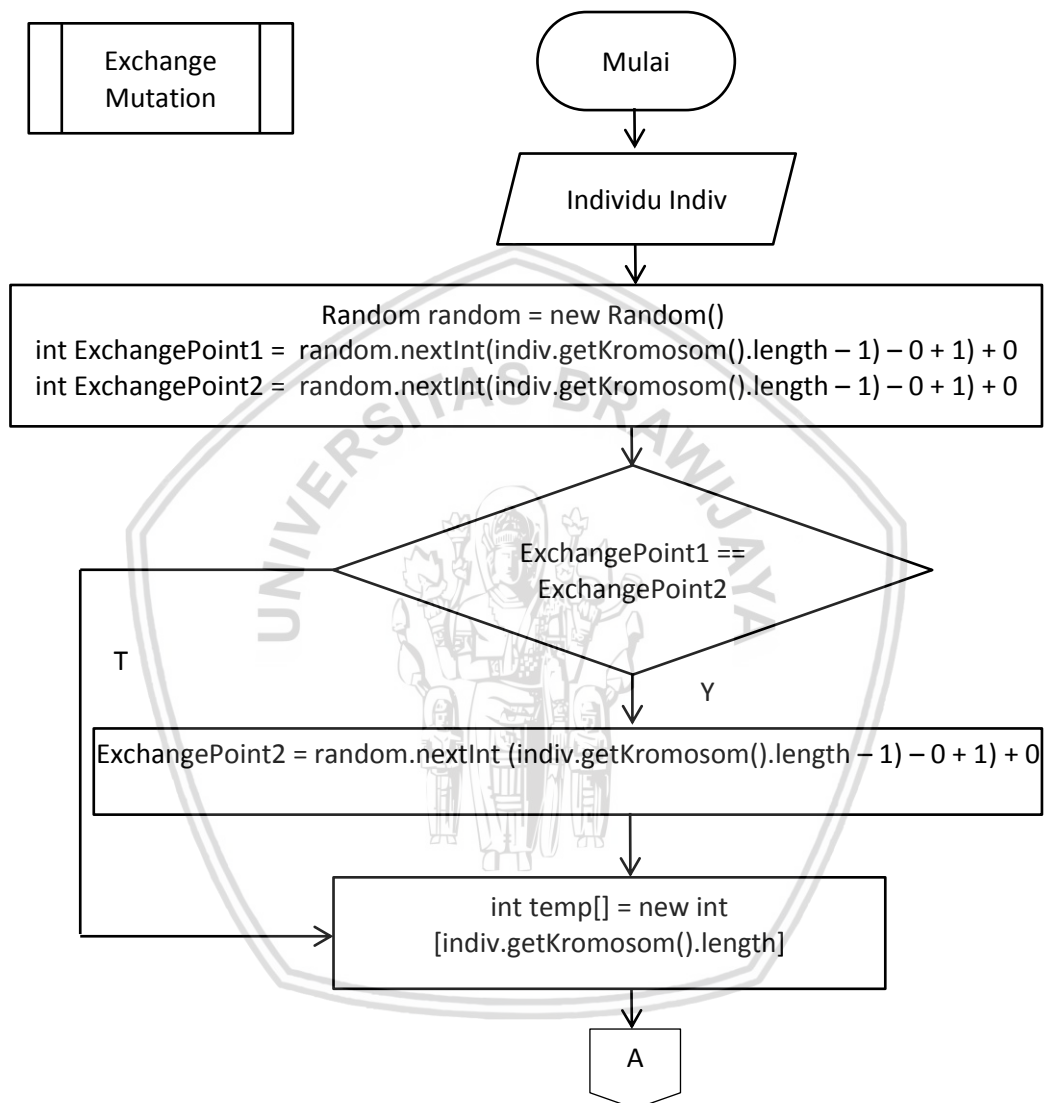
### 4.3.3 Mutasi

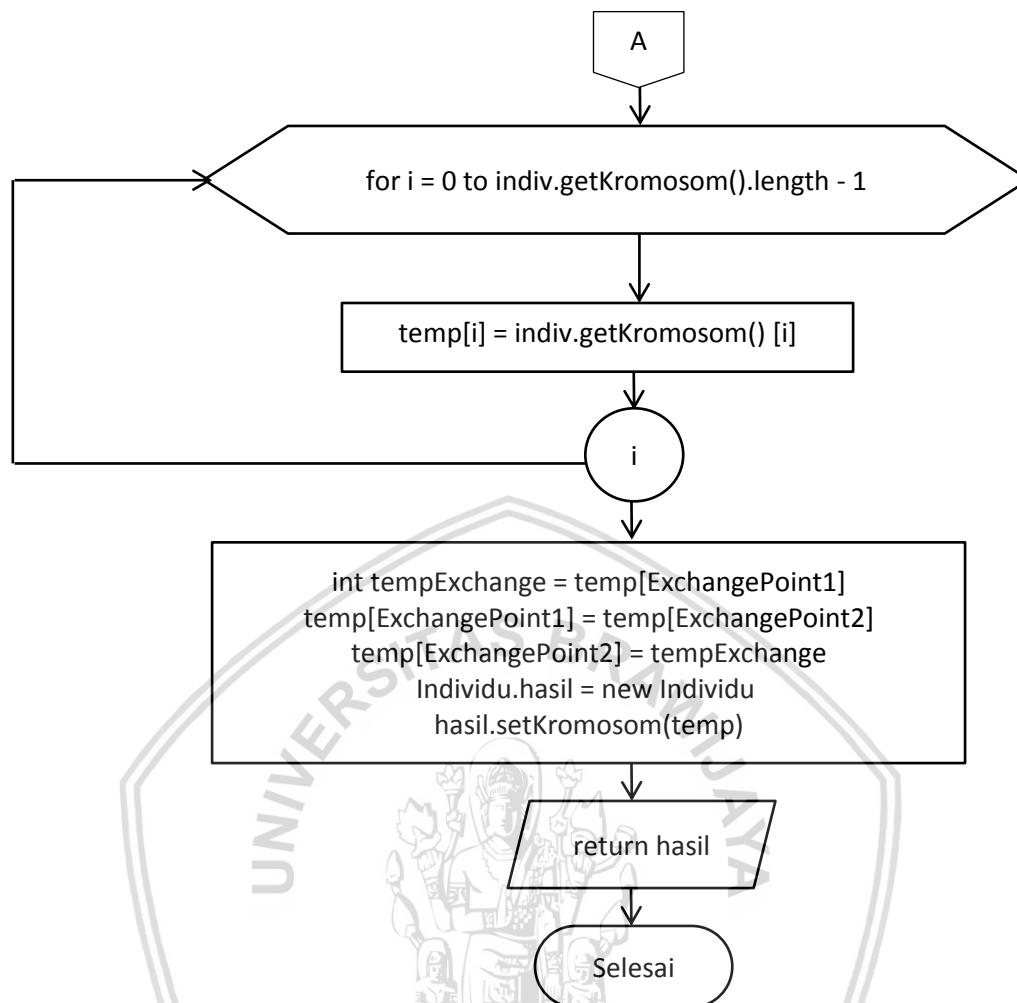
Pada proses mutasi penyelesaian permasalahan yang digunakan adalah menggunakan metode *exchange mutation*. Diagram alir proses mutasi dapat dilihat pada Gambar 4.8.



Gambar 4.8 Diagram Alir Mutasi

Gambar 4.8 merupakan proses reproduksi kedua setelah crossover yaitu proses mutasi. Langkah pertama pada proses ini diawali dengan memasukkan nilai *mr*, *popsiz*e dan populasi. Proses ini dilakukan dengan cara menentukan jumlah *offspring*. Selanjutnya melakukan perulangan sebanyak panjang kromosom mutasi yang dimulai dari 0, kemudian mengambil satu individu secara acak untuk dimutasi sehingga menghasilkan *offspring*.





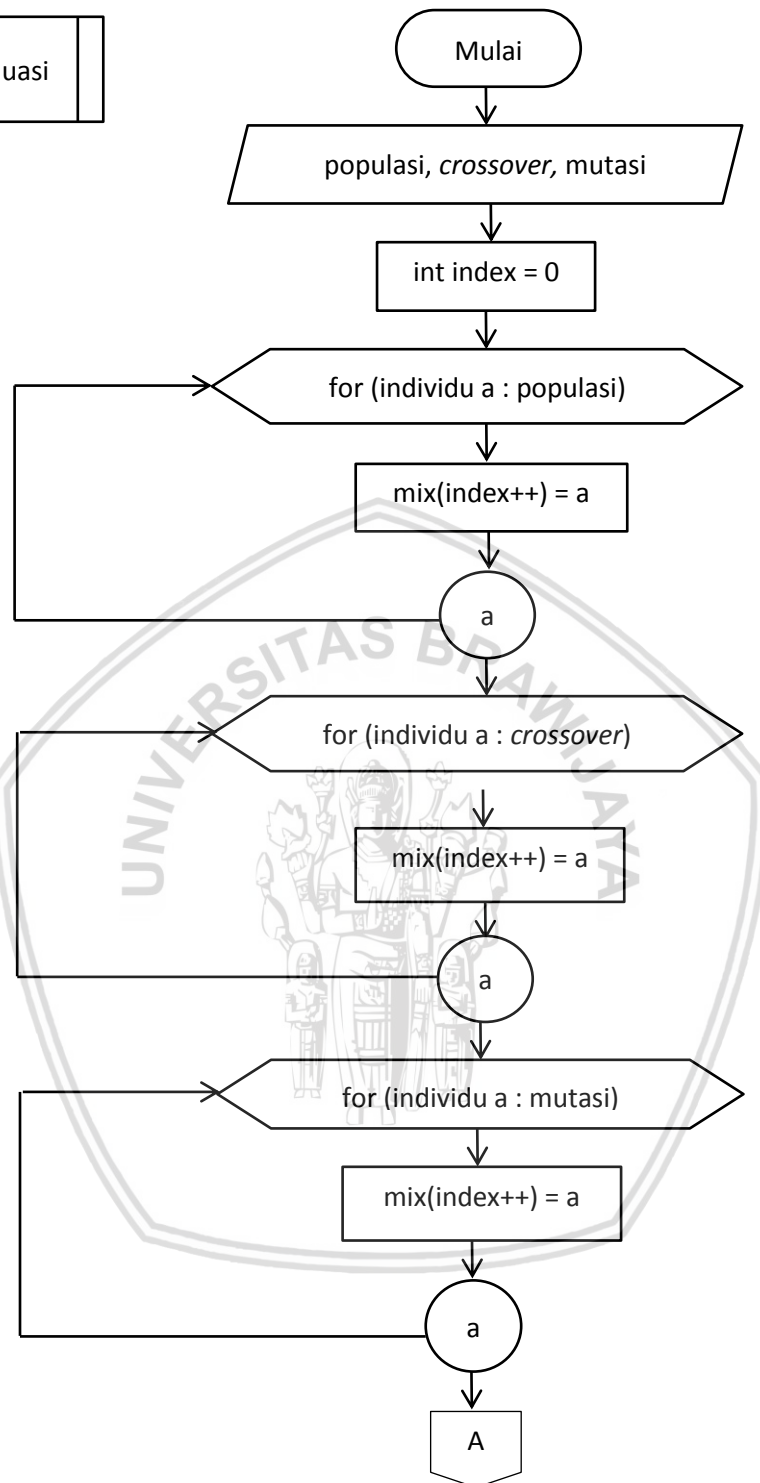
**Gambar 4.9 Diagram Alir *Exchange Mutation***

Pada Gambar 4.9 merupakan proses *exchange mutation* yang diawali dengan menginputkan Individu *Indiv*. Individu *indiv* merupakan *parent* yang sebelumnya sudah dipilih secara acak. Proses ini dilakukan dengan cara menukar individu-individu pada titik tukar yang ditentukan secara acak, sebelumnya akan dipastikan terlebih dahulu agar titik tukar yang ditentukan berbeda dan untuk panjang individu pada kedua titik akan ditukar sehingga menghasilkan *offspring*.

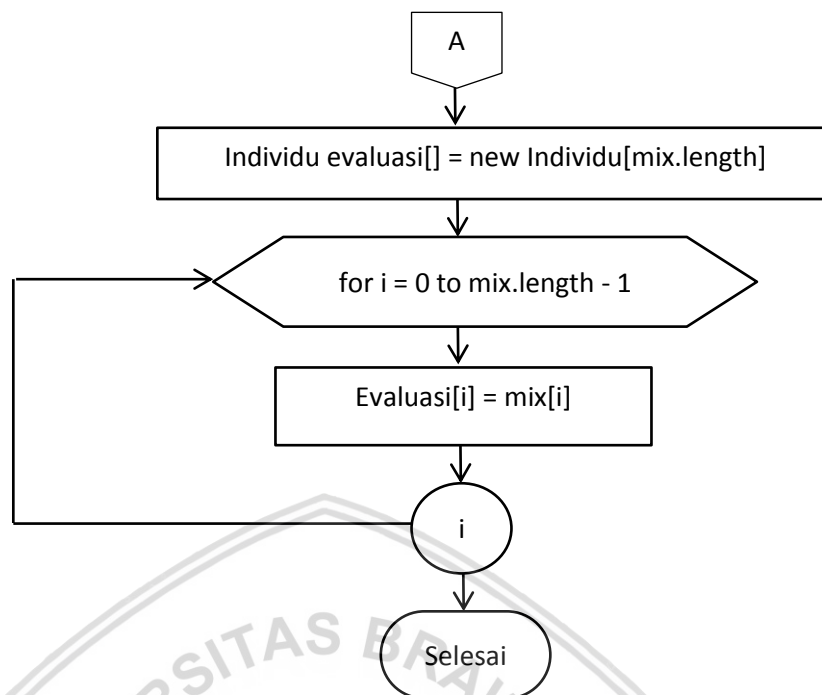
#### 4.3.4 Evaluasi

Evaluasi merupakan prsoses tahapan ketiga dari tahapan-tahapan algoritme genetika. Proses ini dilakukan dengan cara menggabungkan seluruh *parent* dan *child* yang dihasilkan dari masing-masing proses yaitu proses inisialisasi awal, *crossover* dan mutasi. Diagram alir proses evaluasi dapat dilihat pada Gambar 4.10.

Evaluasi
----------





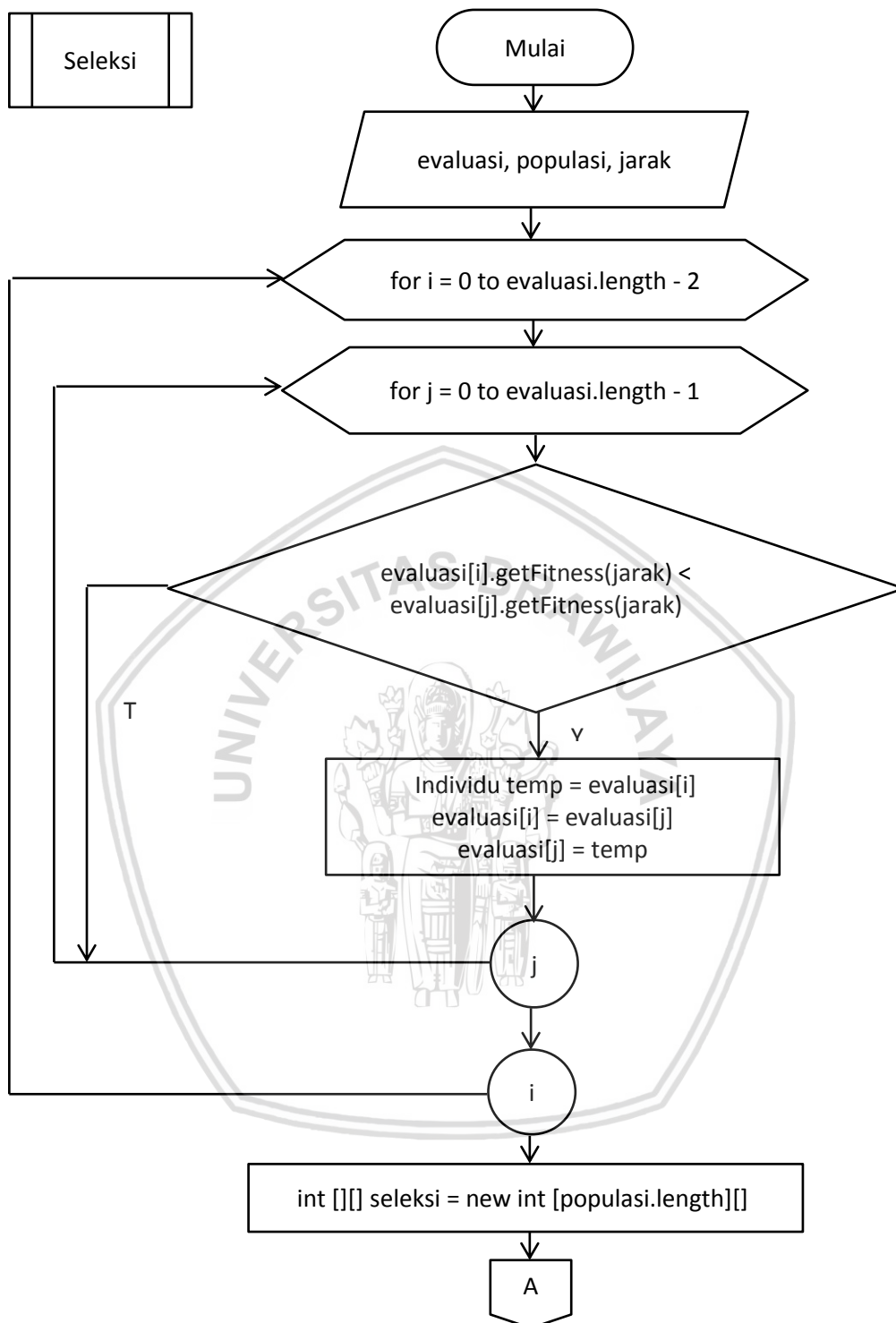


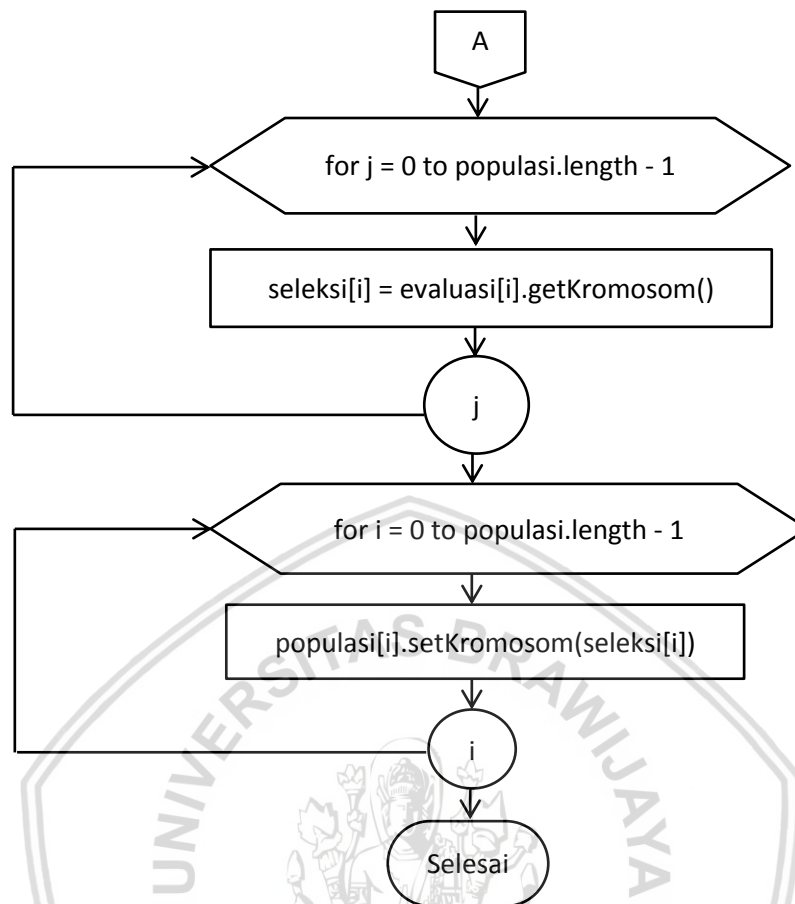
**Gambar 4.10 Diagram Alir Evaluasi**

Pada Gambar 4.10 terdapat tiga inputan yaitu pertama populasi yang merupakan *parent* dari hasil proses inialisasi, kedua *crossover* yang merupakan *child* dari hasil proses *crossover*, dan yang terakhir mutasi merupakan *child* dari hasil proses mutasi. Selanjutnya, akan dilakukan penggabungan seluruh populasi yang ada didalam ketiga variabel tersebut ke dalam *array mix* yang menampung gabungan seluruh populasi *parent* dan *child*, pertama akan dilakukan perulangan untuk memasukkan populasi *parent* pada variabel populasi kedalam variabel *array mix*. Kemudian kembali melakukan perulangan yang sama pada variabel *crossover* untuk menggabungkan populasi *child* di dalamnya ke variabel *array mix*. Yang terakhir adalah perulangan pada variabel mutasi untuk menggabungkan pula populasi *child* di dalamnya ke variabel *array mix*. Setelah seluruh populasi tergabung pada variabel *array mix*. Setelah itu nilai pada *array mix* dipindah ke dalam *array evaluasi*.

#### 4.3.5 Seleksi

Proses seleksi merupakan tahapan terakhir dari tahapan algoritme genetika. Metode seleksi yang digunakan untuk menyelesaikan permasalahan ini menggunakan metode *elitism selection*. Diagram alir seleksi dapat dilihat pada Gambar 4.11.



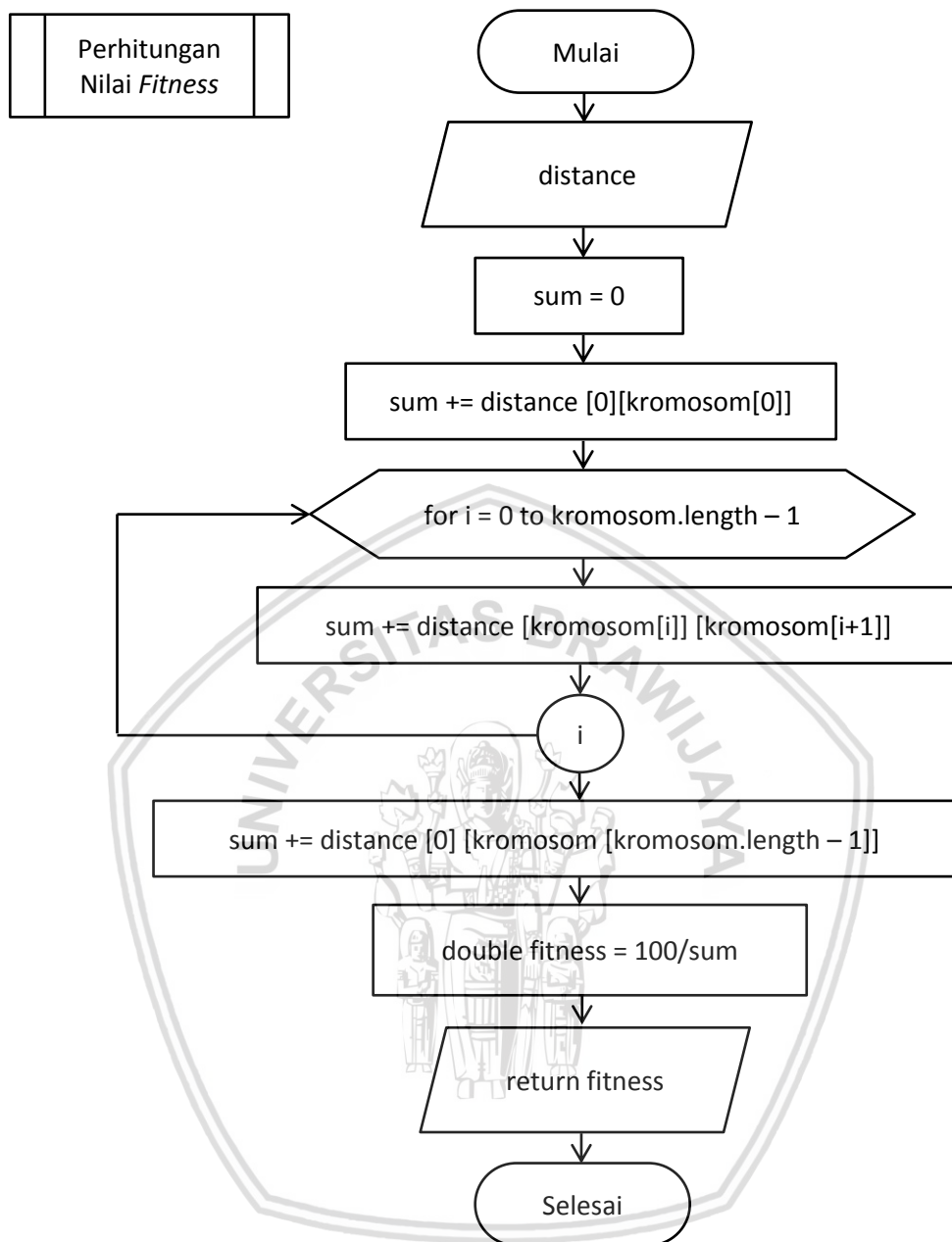


**Gambar 4.11 Diagram Alir Seleksi**

Gambar 4.11 merupakan proses seleksi. Seleksi merupakan proses tahapan terakhir dari tahapan algoritme genetika. Proses ini dilakukan dengan cara mengurutkan nilai fitness dari yang terbesar ke terkecil. Inputan pada proses seleksi yaitu evaluasi, populasi dan jarak. Selanjutnya akan melakukan proses perulangan untuk mengurutkan masing-masing populasi berdasarkan nilai *fitness*. Dari hasil pengurutan tersebut akan dipilih populasi dengan nilai *fitness* terbesar sebanyak jumlah *popsiz*, kemudian merubah kromosom populasi dengan kromosom hasil seleksi. Populasi ini akan digunakan ke dalam generasi selanjutnya sehingga di dapatkan populasi terbaik.

#### 4.3.6 Perhitungan Nilai *Fitness*

Perhitungan nilai *fitness* merupakan proses untuk menghitung nilai *fitness*. Diagram alir perhitungan nilai *fitness* dapat dilihat pada Gambar 4.12.



**Gambar 4.12 Diagram Alir Perhitungan Nilai *Fitness***

Gambar 4.12 merupakan proses perhitungan nilai *fitness*. Proses ini dilakukan dengan cara memasukkan nilai jarak dari parameter lalu membuat variabel *sum* untuk menyimpan total jarak. Selanjutnya menghitung total jarak dari sekolah ke titik awal sampai titik akhir ke sekolah, setelah itu menghitung nilai *fitness* dengan rumus  $100/\text{sum}$  lalu mengembalikan nilai *fitness*.

## 4.4 Perhitungan Manualisasi

### 4.4.1 Inisialisasi Parameter Algoritme Genetika

Pada proses inisialisasi terdapat parameter-parameter algoritme genetika antara lain jumlah generasi, *popsize*, *crossover rate* dan *mutation rate*. Pada penelitian ini parameter yang digunakan untuk menyelesaikan permasalahan pada algoritme genetika adalah sebagai berikut:

- Jumlah Generasi : 1
- *Popsize* : 3
- *Crossover rate* : 0,3
- *Mutation rate* : 0,2

### 4.4.2 Representasi Kromosom

Pada penelitian ini data yang digunakan diambil dari data siswa sekolah MI Salafiyah Kasim yang berada di Blitar. Pengambilan data siswa dilakukan dengan melakukan observasi langsung dengan membuat janji pertemuan dan wawancara dengan pihak sekolah. Jumlah data siswa yang didapatkan dari pihak sekolah sebanyak 44 jumlah siswa yang menaiki angkutan bus sekolah saat pengantaran siswa. Sistem pengantaran siswa dibagi menjadi dua kloter yaitu kloter pagi dan kloter siang. Pada kloter pagi pengantaran siswa mulai diberangkatkan pada pukul 11.30 WIB, sedangkan pada kloter siang pengantaran siswa mulai diberangkatkan pada pukul 13.00 WIB. Jumlah siswa kloter pagi sebanyak 20 siswa dan kloter siang sebanyak 24 siswa. Pengantaran siswa telah diasumsikan di persimpangan jalan. Siswa yang menaiki bus angkutan sekolah merupakan siswa yang rumahnya jauh dari sekolah dengan jarak sekitar satu sampai dua km dari sekolah, dan siswa yang menaiki angkutan sekolah tidak selalu sama setiap harinya dikarenakan ada siswa yang sakit, ada siswa yang dijemput orang tuanya dan lain-lain.

Panjang representasi kromosom yang digunakan untuk menyelesaikan contoh permasalahan ini sebanyak 38 siswa yang terbagi menjadi dua kloter yaitu kloter pagi dan kloter siang. Pada kloter pagi sebanyak 18 siswa dan kloter siang sebanyak 20 siswa. Berikut Contoh panjang representasi kromosom yang digunakan untuk menyelesaikan contoh permasalahan dapat dilihat pada Tabel 4.3 da Tabel 4.4.

**Tabel 4.3 Representasi Kromosom Pagi**

Representasi Kromosom																		
Individu	PAGI																	
	S	19	18	17	16	15	14	13	10	9	8	7	6	5	4	3	2	S
P1	S	3	5	7	9	13	15	17	19	2	4	6	8	10	14	16	18	S
P2	S	10	14	8	6	4	2	18	7	19	16	15	9	17	5	13	3	S
P3	S	6	2	16	14	7	3	9	5	17	13	19	10	4	18	15	8	S

**Tabel 4.4 Representasi Kromosom Siang**

		Representasi Kromosom																		
Individu	SIANG																			
	S	1	2	4	5	6	9	10	11	12	13	14	17	19	20	21	22	23	24	S
P1	S	23	21	19	17	13	11	9	5	1	24	22	20	14	12	10	6	4	2	S
P2	S	1	14	9	5	20	21	1	23	2	24	4	22	6	13	12	19	10	17	S
P3	S	6	20	24	2	13	4	19	9	22	5	14	21	11	23	17	10	12	1	S

#### 4.4.3 Inisialisasi Populasi Awal

Pada contoh permasalahan berikut ditentukan jumlah *popsiz* sebanyak 3, jumlah hari 1 yaitu hari sabtu terbagi menjadi kloter pagi dan siang sehingga panjang kromosom sebanyak  $3 \times 34 = 102$ . Contoh permasalahan pada inisialisasi awal dapat dilihat pada Tabel 4.5 dan Tabel 4.6.



**Tabel 4.5 Inisialisasi Awal Pagi**

		Kromosom																			
Individu		PAGI																		Total Jarak	Fitness
P1	S	3	5	7	9	13	15	17	19	2	4	6	8	10	14	16	18	S			
		1.6	0.1	1.6	3.5	1.4	3.4	0.9	4	5.6	0.2	0.1	1.5	2.3	2.3	1	0.9	1	31.4	3.184713376	
P2	S	10	14	8	6	4	2	18	7	19	16	15	9	17	5	13	3	S			
		1.3	2.3	2.2	1.4	0.1	0.4	2.4	2.3	5.5	3.9	0.8	2.4	1.4	2.1	4.6	4.6	1.6	39.3	2.544529262	
P3	S	6	2	16	14	7	3	9	5	17	13	19	10	4	18	15	8	S			
		1.9	0.4	1.8	1	2.1	1.6	3.3	3.3	2.1	2.8	6.5	5.3	0.8	2.9	0.2	2.1	1.9	40	2.5	

**Tabel 4.6 Inisialisasi Awal Siang**

Kromosom																						
Individu	PAGI																				Total Jarak	Fitness
P1	S	23	21	19	17	13	11	9	5	1	24	22	20	14	12	10	6	4	2	S		
		3.1	8.7	6.2	1.5	1.2	1.1	0.5	4.9	5.7	1.9	2.5	2.8	4.8	5.2	0.4	4.4	0.4	4.7	1.3	61.3	1.63132137
P2	S	11	14	9	5	20	21	1	23	2	24	4	22	6	13	12	19	10	17	S		

		1.4	5.2	6.7	4.9	7	8.8	8	2.2	2.3	2	6.6	4.2	3.9	3.8	1.1	0.3	0.3	1.8	0.5	71	1.408450704
<b>P3</b>	S	6	20	24	2	13	4	19	9	22	5	14	21	11	23	17	10	12	1	S		
		3.1	6.7	7	2	2.5	4.1	4.6	0.5	2.3	4.2	8.8	10	6.3	2.3	3.6	1.8	0.4	2	1.4	73.6	1.358695652

#### 4.4.4 Crossover

Proses *crossover* yang digunakan pada permasalahan menggunakan *one cut point*. Sebelum melakukan proses *crossover* terlebih dahulu untuk menentukan berapa jumlah *offspring* yang akan dihasilkan. Cara untuk mendapatkan hasil *offspring* dari *crossover* yaitu dengan mengkalikan *crossover rate* dengan *popsiz*:

$$\text{offspring} = 0,3 \times 3 = 0,9 \text{ (dibulatkan menjadi 1)}$$

Dari perhitungan diatas, maka jumlah *offspring* yang akan dihasilkan sebanyak 2 *offspring*. Tahap selanjutnya adalah memilih dua *parent* yang dilakukan secara *random* dalam proses inisialisasi awal. Misalkan 2 *parent* yang diambil adalah P2 dan P3 maka proses *crossover* dapat dilihat pada Tabel 4.7 dan Tabel 4.8.

**Tabel 4.7 Crossover Pagi**

PAGI																		
<b>P2</b>	S	10	14	8	6	4	2	18	7	19	16	15	9	17	5	13	3	S
<b>P3</b>	S	6	2	16	14	7	3	9	5	17	13	19	10	4	18	15	8	S

Cut Point di titik ke-6

<b>C1</b>	S	10	14	8	6	4	2	16	7	3	9	5	17	13	19	18	15	S	<b>Total Jarak</b>	<b>Fitness</b>
-----------	---	----	----	---	---	---	---	----	---	---	---	---	----	----	----	----	----	---	--------------------	----------------

		1.3	2.3	2.2	1.5	0.1	0.4	1.8	2.3	1.6	3.3	3.3	2.1	2.8	6.5	3.6	0.2	0.8	36.1	2.770083102
--	--	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	-------------

**Tabel 4.8 Crossover Siang**

SIANG																				
<b>P2</b>	S	11	14	9	5	20	21	1	23	2	24	4	22	6	13	12	19	10	17	S
<b>P3</b>	S	6	20	24	2	13	4	19	9	22	5	14	21	11	23	17	10	12	1	S

Cut Point di titik ke-12

<b>C1</b>	S	11	14	9	5	20	21	1	23	2	24	4	22	6	13	19	17	10	12	S	Total Jarak	Fitness
		1.4	5.2	6.7	4.9	7	8.8	8	2.2	2.3	2	6.6	4.2	3.9	3.8	1.5	1.5	1.8	0.4	1.3	73.5	1.360544218

Berdasarkan hasil *crossover* diatas didapatkan dua *offspring* yaitu C1 dan C2. *Offspring* hasil *crossover* yang didapatkan dapat dilihat pada Tabel 4.9 dan Tabe 4.10:

**Tabel 4.9 Hasil *Offspring* dari Kloter Pagi**

PAGI																				
<b>C1</b>	S	10	14	8	6	4	2	16	7	3	9	5	17	13	19	18	15	S	Total Jarak	Fitness
		1.3	2.3	2.2	1.5	0.1	0.4	1.8	2.3	1.6	3.3	3.3	2.1	2.8	6.5	3.6	0.2	0.8	36.1	2.770083102

**Tabel 4.10 Hasil *Offspring* dari Kloter Siang**

SIANG																						
C1	S	11	14	9	5	20	21	1	23	2	24	4	22	6	13	19	17	10	12	S	Total Jarak	Fitness
		1.4	5.2	6.7	4.9	7	8.8	8	2.2	2.3	2	6.6	4.2	3.9	3.8	1.5	1.5	1.8	0.4	1.3	73.5	1.360544218

#### 4.4.5 Mutasi

Proses mutasi yang digunakan pada permasalahan ini adalah metode *exchange mutation*. Proses ini dilakukan dengan cara menukar dua titik yang akan dimutasi lalu nilainya ditukar. Cara untuk menghitung *mutation* yaitu dengan mengkalikan *mutation rate* dengan *popsiz*.

$$offspring = 0,2 \times 3 = 0,6 \text{ (dibulatkan menjadi 1)}$$

Dari proses perhitungan diatas, maka jumlah *offspring* yang akan dihasilkan dari *mutation* sebanyak 2 *offspring*. Selanjutnya memilih parent yang dilakukan secara acak dari populasi awal. Misalkan parent yang yang dipilih adalah P1 maka proses mutasi dapat dilihat pada Tabel 4.11 dan Tabel 4.12.

**Tabel 4.11 Mutasi Pagi**

PAGI																		
P1	S	3	5	7	9	13	15	17	19	2	4	6	8	10	14	16	18	S

Titik Potong di titik ke 3 dan 8

C2	S	3	5	7	9	13	15	17	19	2	4	6	8	10	14	16	18	S	Total Jarak	Fitness
		1.6	0.1	6.3	5.1	1.4	3.4	0.9	2.5	1.8	0.2	0.1	1.5	2.3	2.3	1	0.9	1	32.4	3.086419753

Tabel 4.12 Mutasi Siang

SIANG																				
P1	S	23	21	19	17	13	11	9	5	1	24	22	20	14	12	10	6	4	2	S

Titik Potong di titik ke 11 dan 17

C2	S	23	21	19	17	13	11	9	5	1	24	4	20	14	12	10	6	22	2	S	Total Jarak	Fitness
		3.1	8.7	6.2	1.5	1.2	1.1	0.7	4.9	5.7	1.9	6.6	7	4.8	5.2	0.4	4.4	3.9	0.6	1.3	69.2	1.445086705

Berdasarkan hasil mutasi diatas didapatkan dua *offspring* yaitu C3 dan C4. *Offspring* hasil *crossover* yang didapatkan dapat dilihat pada Tabel 4.13 dan Tabe 4.14:

**Tabel 4.13 Hasil *Offspring* dari Kloter Pagi**

PAGI																				
C2	S	3	5	7	9	13	15	17	19	2	4	6	8	10	14	16	18	S	Total Jarak	Fitness
		1.6	0.1	6.3	5.1	1.4	3.4	0.9	2.5	1.8	0.2	0.1	1.5	2.3	2.3	1	0.9	1	32.4	3.086419753

**Tabel 4.14 Hasil *Offspring* dari Kloter Siang**

		SIANG																				
C2	S	23	21	19	17	13	11	9	5	1	24	4	20	14	12	10	6	22	2	S	Total Jarak	Fitness
		3.1	8.7	6.2	1.5	1.2	1.1	0.7	4.9	5.7	1.9	6.6	7	4.8	5.2	0.4	4.4	3.9	0.6	1.3	69.2	1.445086705

#### 4.4.6 Evaluasi

Setelah proses *crossover* dan mutasi selesai maka didapatkan *offspring*. Kemudian *offspring* hasil dari *crossover* dan mutasi tersebut akan digabungkan menjadi satu. Nilai *fitness* dari tiap individu didapatkan pada saat melakukan proses evaluasi, sehingga akan didapatkan nilai *fitness* yang terbaik. Penggabungan dari populasi dapat dilihat pada Tabel 4.15 dan Tabel 4.16.



**Tabel 4.15 Hasil Populasi Baru Kloter Pagi**

Kromosom																			
Individu	PAGI																		Fitness
<b>P1</b>	S	3	5	7	9	13	15	17	19	2	4	6	8	10	14	16	18	S	3.184713376
<b>P2</b>	S	10	14	8	6	4	2	18	7	19	16	15	9	17	5	13	3	S	2.544529262
<b>P3</b>	S	6	2	16	14	7	3	9	5	17	13	19	10	4	18	15	8	S	2.5
<b>C1</b>	S	10	14	8	6	4	2	16	7	3	9	5	17	13	19	18	15	S	2.770083102
<b>C2</b>	S	3	5	7	9	13	15	17	19	2	4	6	8	10	14	16	18	S	3.086419753

**Tabel 4.16 Hasil Populasi Baru Kloter Siang**

		Kromosom																			
Individu		SIANG																			Fitness
P1	S	23	21	19	17	13	11	9	5	1	24	22	20	14	12	10	6	4	2	S	1.63132137
P2	S	1	14	9	5	20	21	1	23	2	24	4	22	6	13	12	19	10	17	S	1.408450704
P3	S	6	20	24	2	13	4	19	9	22	5	14	21	11	23	17	10	12	1	S	1.358695652
C1	S	11	14	9	5	20	21	1	23	2	24	4	22	6	13	19	17	10	12	S	1.360544218
C2	S	23	21	19	17	13	11	9	5	1	24	4	20	14	12	10	6	22	2	S	1.445086705

### 1.1.1.1 Proses Perhitungan Nilai *Fitness*

Proses perhitungan *fitness* dilakukan dengan tujuan untuk mencari nilai terbaik dari masing-masing individu. Sebelum menghitung nilai *fitness* terlebih dahulu menentukan total jarak yang dilewati masing-masing individu, nilai *fitness* dihitung dengan rumus dibawah ini:

$$\text{Nilai } fitness = \frac{100}{\text{Total Jarak yang dilewati}}$$

Proses perhitungan nilai *fitness* populasi baru yang berasal dari proses evaluasi dapat dilihat pada Tabel 4.17 dan Tabel 4.18.

**Tabel 4.17 Hasil Perhitungan Nilai *Fitness* Kloter Pagi**

Kromosom																				
Individu	PAGI																		Total Jarak	Fitness
P1	S	3	5	7	9	13	15	17	19	2	4	6	8	10	14	16	18	S	31.4	3.184713376
P2	S	10	14	8	6	4	2	18	7	19	16	15	9	17	5	13	3	S	39.3	2.544529262
P3	S	6	2	16	14	7	3	9	5	17	13	19	10	4	18	15	8	S	40	2.5
C1	S	10	14	8	6	4	2	16	7	3	9	5	17	13	19	18	15	S	36.1	2.770083102
C2	S	3	5	7	9	13	15	17	19	2	4	6	8	10	14	16	18	S	32.4	3.086419753

**Tabel 4.18 Hasil Perhitungan Nilai *Fitness* Kloter Siang**

Kromosom																				
Individu	SIANG																		Total Jarak	Fitness

<b>P1</b>	S	23	21	19	17	13	11	9	5	1	24	22	20	14	12	10	6	4	2	S	61.3	1.63132137
<b>P2</b>	S	1	14	9	5	20	21	1	23	2	24	4	22	6	13	12	19	10	17	S	71	1.408450704
<b>P3</b>	S	6	20	24	2	13	4	19	9	22	5	14	21	11	23	17	10	12	1	S	73.6	1.358695652
<b>C1</b>	S	11	14	9	5	20	21	1	23	2	24	4	22	6	13	19	17	10	12	S	73.5	1.360544218
<b>C2</b>	S	23	21	19	17	13	11	9	5	1	24	4	20	14	12	10	6	22	2	S	69.2	1.445086705

#### 4.4.7 Seleksi

Proses seleksi yang digunakan pada permasalahan ini adalah *elitism selection* yang bertujuan untuk melakukan penggabungan seluruh nilai *fitness* sehingga dapat diurutkan berdasarkan nilai tertinggi hingga nilai terendah. Selanjutnya akan dipilih individu yang memiliki nilai *fitness* tertinggi sebanyak jumlah *popsiz* yaitu ada tiga individu yang dipilih. Individu yang terpilih akan digunakan sebagai parent pada generasi selanjutnya. Proses seleksi dapat dilihat pada Tabel 4.19 dan Tabel 4.20.

**Tabel 4.19 Hasil Seleksi Kloter Pagi**

PAGI			
Individu	Total Nilai <i>Fitness</i>	Sorting	Nilai <i>Fitness</i>
P1	3.184713376	P1	3.184713376
P2	2.544529262	C2	3.086419753
P3	2.5	C1	2.770083102
C1	2.770083102		
C2	3.086419753		

Tabel 4.20 Hasil Seleksi Kloter Siang

SIANG			
Individu	Total Nilai <i>Fitness</i>	Sorting	Nilai <i>Fitness</i>
P1	1.63132137	P3	1.63132137
P2	1.408450704	C2	1.445086705
P3	1.358695652	C1	1.408450704
C1	1.360544218		
C2	1.445086705		

## 4.5 Perancangan *User Interface*

Berdasarkan rancangan *user interface* (antarmuka) yang telah diuraikan, sistem ini akan menggunakan GUI Java yang terbagi menjadi beberapa komponen yaitu antarmuka halaman utama, antarmuka data siswa, antarmuka halaman populasi awal, antarmuka proses algoritme genetika dan antarmuka output hasil algoritme genetika.

### 4.5.1 Perancangan *User Interface* Halaman Utama

Pada perancangan antarmuka halaman utama terdapat beberapa halaman yaitu halaman inputan, halaman proses algoritme genetika dan halaman hasil rute. Perancangan halaman utama dapat dilihat pada Gambar 4.12

**SISTEM OPTIMASI TSP MENGGUNAKAN ALGORITME GENETIKA**

Inputan
Proses Algoritme Genetika
Hasil Rute

<div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 1</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 2</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 3</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 4</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 5</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 6</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 7</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 8</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 9</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 10</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 11</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 12</div> </div>	<div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 1</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 2</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 3</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 4</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 5</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 6</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 7</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 8</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 9</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 10</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 11</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 12</div> </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">Anak 13</div> </div>
--	--

**Parameter**

**Popsi**

**Generasi**

Default

**Crossover Rate**

**Mutation Rate**

Proses

**Gambar 4.13 Tampilan Halaman Utama**

#### 4.5.2 Perancangan *User Interface* Halaman Proses Algoritme Genetika

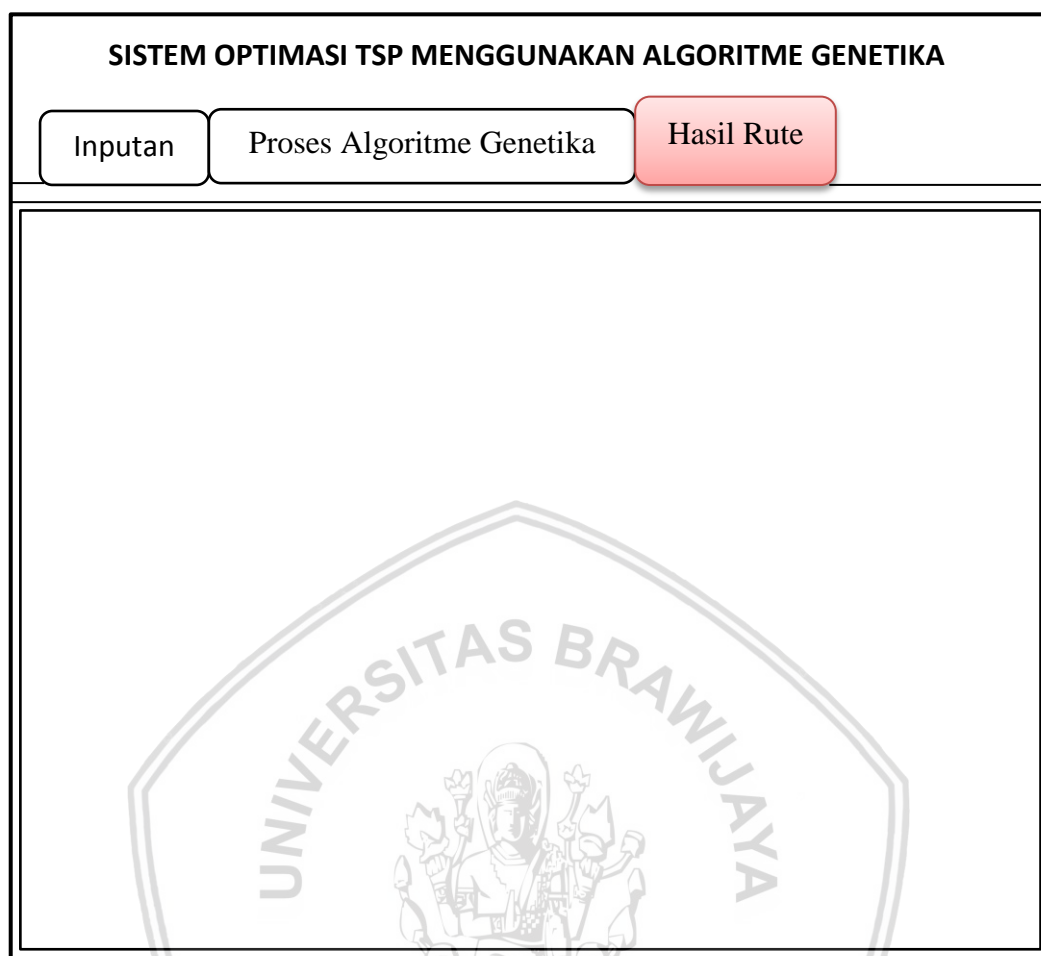
Pada perancangan *user interface* halaman proses algoritme genetika terdiri dari beberapa halaman yaitu halaman populasi awal, halaman *crossover*, halaman mutasi, dan halaman individu terbaik. Halaman ini bertujuan untuk menampilkan hasil proses dari algoritme genetika. Perancangan halaman proses algoritme genetika dapat dilihat pada Gambar 4.13.

<b>SISTEM OPTIMASI TSP MENGGUNAKAN ALGORITME GENETIKA</b>														
Inputan	Proses Algoritme Genetika		Hasil Rute											
<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px 10px;">Kloter Pagi</div> <div style="border: 1px solid black; padding: 5px 10px;">Kloter Siang</div> </div>														
<div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="border: 1px solid black; padding: 5px 10px;">Populasi Awal</div> <div style="border: 1px solid black; padding: 5px 10px;">Crossover</div> <div style="border: 1px solid black; padding: 5px 10px;">Mutasi</div> <div style="border: 1px solid black; padding: 5px 10px;">Evaluasi</div> <div style="border: 1px solid black; padding: 5px 10px;">Sorting</div> <div style="border: 1px solid black; padding: 5px 10px;">Seleksi</div> </div>														
<div style="border: 1px solid black; padding: 5px 20px; margin: 10px auto; width: 60%;">Individu Terbaik</div>														
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Generasi Ke-</th> <th style="width: 15%;">Individu</th> <th style="width: 15%;">Kromosom</th> <th style="width: 15%;">Total Jarak</th> <th style="width: 15%;">Fitness</th> </tr> </thead> <tbody> <tr> <td colspan="5" style="height: 250px; position: relative;"> <div style="position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%); opacity: 0.1; font-size: 100px; pointer-events: none;"> UNIVERSITAS BRAWIJAYA </div> </td> </tr> </tbody> </table>					Generasi Ke-	Individu	Kromosom	Total Jarak	Fitness	<div style="position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%); opacity: 0.1; font-size: 100px; pointer-events: none;"> UNIVERSITAS BRAWIJAYA </div>				
Generasi Ke-	Individu	Kromosom	Total Jarak	Fitness										
<div style="position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%); opacity: 0.1; font-size: 100px; pointer-events: none;"> UNIVERSITAS BRAWIJAYA </div>														

**Gambar 4.14 Halaman Proses Algoritme Genetika**

#### 4.5.3 Perancangan *User Interface* Halaman Hasil Rute

Pada perancangan *user interface* halaman hasil rute akan menampilkan hasil rute dari proses algoritme genetika. Hasil yang muncul berupa urutan nama dan alamat siswa yang akan direkomendasikan kepada supir angkutan sekolah. Perancangan halaman hasil rute dapat dilihat pada Gambar 4.14.



Gambar 4.15 Halaman Hasil Rute

#### 4.5.4 Pengujian Ukuran Populasi

Pengujian ukuran populasi dilakukan untuk mengetahui pengaruh dari jumlah populasi terhadap nilai *fitness* yang dihasilkan. Untuk mendapatkan individu yang lebih beragam dalam menghasilkan solusi terbaik ukuran populasi yang lebih besar maka akan lebih baik jika ukuran populasi semakin besar. Pada pengujian ini ukuran populasi yang digunakan dimulai dari 10 hingga 100 dan dilakukan sebanyak 10 kali. Uji coba ukuran populasi dapat dilihat pada Tabel 4.21.

Tabel 4. 21 Pengujian Ukuran Populasi

Ukuran Populasi	Nilai <i>Fitness</i> Pengujian Ke – <i>i</i>										Nilai <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
10											
20											
30											
40											



50											
60											
70											
80											
90											
100											

#### 4.5.5 Pengujian Kombinasi Nilai *Crossover Rate* dan *Mutation Rate*

Pengujian ini dilakukan dengan tujuan untuk mengetahui nilai optimal yang dihasilkan dari perhitungan yang menggunakan nilai *cr* dan *mr*. Nilai yang digunakan pada parameter *cr* dan *mr* merupakan bilangan integer mulai dari 0,1 hingga 1. Uji coba kombinasi nilai *cr* dan *mr* dapat dilihat pada Tabel 4.22.

Tabel 4. 22 Pengujian Kombinasi Nilai *Cr* dan *Mr*

Kombinasi $C_r ; M_r$	Nilai <i>Fitness</i> Pengujian Ke – $i$										Nilai <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
0,1 ; 1											
0,2 ; 0,9											
0,3 ; 0,8											
0,4 ; 0,7											
0,5 ; 0,6											
0,6 ; 0,5											
0,7 ; 0,4											
0,8 ; 0,3											
0,9 ; 0,2											
1 ; 0,1											

#### 4.5.6 Pengujian Uji Konvergensi

Pengujian ini dilakukan dengan tujuan untuk menemukan nilai konvergen dari rata-rata nilai *fitness* pada setiap generasi yang di proses dalam algoritme genetika. Uji coba konvergensi dapat dilihat pada Tabel 4.23.

Tabel 4.23 Pengujian Uji Konvergensi

Banyak Generasi	Nilai <i>Fitness</i> Pengujian Ke – <i>i</i>					Nilai Generasi Ketika Konvergen
	1	2	3	4	5	
1						
2						
4						
6						
8						
10						
30						
60						
90						
120						
250						
500						
1000						

## BAB 5 IMPLEMENTASI

Pada bab ini akan dijelaskan mengenai proses implementasi berdasarkan analisis kebutuhan dan perancangan sistem yang telah dijelaskan sebelumnya. Selain itu juga akan digambarkan *user interface* dari sistem yang telah dibangun.

### 5.1 Spesifikasi Sistem

Pada proses pengimplementasian dilakukan dengan membangun sistem berdasarkan hasil analisis kebutuhan dan perancangan sistem yang dibutuhkan. Spesifikasi perangkat yang digunakan terdiri dari 2 macam yaitu *hardware* dan *software*.

#### 5.1.1 Spesifikasi Perangkat Keras

Spesifikasi hardware yang digunakan dalam penyelesaian permasalahan implementasi algoritma genetika untuk optimasi *travelling salesman problem* pada angkutan sekolah dapat dilihat pada Tabel 5.1.

**Tabel 5.1 Spesifikasi Perangkat Keras**

Komponen	Spesifikasi
Prosesor	Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz 2.19 GHz
Memori	4 GB
Hardisk	500 GB

#### 5.1.2 Spesifikasi Perangkat Lunak

Spesifikasi software yang digunakan dalam penyelesaian permasalahan implementasi algoritma genetika untuk optimasi *travelling salesman problem* pada angkutan sekolah dapat dilihat pada Tabel 5.2.

**Tabel 5.2 Spesifikasi Lunak**

Komponen	Spesifikasi
Sistem operasi	Windows 8
Bahasa Pemograman	Java
Tools Pemograman	NetBeans IDE

### 5.2 Implementasi Program

Spesifikasi sistem dituliskan dalam bentuk kode program yang menjelaskan alur sistem yang menggunakan algoritme genetika. Sistem ini bertujuan untuk melakukan optimasi rute pada angkutan sekolah berdasarkan jarak.

### 5.2.1 Implementasi Proses Inisialisasi Populasi Awal

Proses inisialisasi populasi awal dilakukan dengan membangkitkan populasi secara *random*. Jumlah populasi yang terdapat di dalamnya telah ditentukan sebelumnya sebanyak jumlah *popsiz*, selain itu jumlah panjang kromosom didapatkan dari banyaknya jumlah siswa yang dipilih oleh pengguna sistem. Proses inisialisasi awal dapat dilihat pada Kode Program 5.1.

Baris	Kode Program
1	<code>int[] Kromosom1 = new int[IDselected1.length];</code>
2	<code>for (int i = 0; i &lt; Kromosom1.length; i++) {</code>
3	<code>    Kromosom1[i] = IDselected1[i] + 1;</code>
4	<code>}</code>
5	<code>int[] Kromosom2 = new int[IDselected2.length];</code>
6	<code>for (int i = 0; i &lt; Kromosom2.length; i++) {</code>
7	<code>    Kromosom2[i] = IDselected2[i] + 1;</code>
8	<code>}</code>
9	
10	<code>Individu populasi1[] = new Individu[popSize];</code>
11	<code>String popString1[] = new String[popSize];</code>
12	<code>for (int i = 0; i &lt; populasi1.length; i++) {</code>
13	<code>    populasi1[i] = new Individu(Kromosom1);</code>
14	<code>    popString1[i] = populasi1[i].toString();</code>
15	<code>}</code>
16	
17	<code>for (int i = 0; i &lt; populasi1.length; i++) {</code>
18	<code>    populasi1[i].setKromosom(popString1[i]);</code>
19	<code>}</code>
20	
21	
22	<code>Individu populasi2[] = new Individu[popSize];</code>
23	<code>String popString2[] = new String[popSize];</code>
24	<code>for (int i = 0; i &lt; populasi2.length; i++) {</code>
25	<code>    populasi2[i] = new Individu(Kromosom2);</code>
26	<code>    popString2[i] = populasi2[i].toString();</code>
27	<code>}</code>
28	
29	<code>for (int i = 0; i &lt; populasi2.length; i++) {</code>
30	<code>    populasi2[i].setKromosom(popString2[i]);</code>
31	<code>}</code>

**Kode Program 5.1 Implementasi Proses Inisialisasi Awal**

Berikut penjelasan Kode Program 5.1:

1. Baris 1, Inisialisasi variabel *array* kromosom sebanyak jumlah ID yang di pilih.
2. Baris 2-4, Perulangan untuk menentukan nilai dari gen kromosom berdasarkan ID yang dipilih.
3. Baris 5, Inisialisasi variabel *array* kromosom2 (siang) sebanyak jumlah ID yang di pilih.

4. Baris 6, Perulangan untuk menentukan nilai dari gen kromosom2 berdasarkan ID yang dipilih.
5. Baris 10, Inisialisasi variabel *array* populasi1 sebanyak jumlah *popsi*.
6. Baris 11, Inisialisasi variabel *array* popString1 sebanyak jumlah *popsi*.
7. Baris 12-15, Perulangan untuk menentukan nilai populasi1 ke i dengan memanggil *constructor*-nya dengan parameter kromosom1 dan menentukan nilai popString1 i dengan nilai method toString populasi1 ke – i.
8. Baris 17-19, Perulangan untuk menentukan nilai populasi1 ke i dengan memanggil *method* setKromosom dengan parameter popString1 ke – i.
9. Baris 22, Inisialisasi variabel *array* populasi2 sebanyak jumlah *popsi*.
10. Baris 23, Inisialisasi variabel *array* popString2 sebanyak jumlah *popsi*.
11. Baris 24-27, Perulangan untuk menentukan nilai populasi2 ke – i dengan memanggil *constructor*-nya dengan parameter kromosom2 dan menentukan nilai popString2 i dengan nilai *method* toString populasi2 ke – i.
12. Baris 29-31, Perulangan untuk menentukan nilai populasi2 ke i dengan memanggil *method* setKromosom dengan parameter popString2 ke - i.

### 5.2.2 Implementasi Proses Crossover

Setelah melakukan proses inisialisasi dilakukan proses reproduksi. Proses reproduksi pertama yang dilakukan dalam algoritme genetika adalah *crossover*. Proses ini dilakukan dengan cara mengambil *parent* secara *random* dari proses hasil inisialisasi, kemudian menentukan titik potong pada masing-masing *parent* untuk melakukan pertukaran gen. Selain itu juga dilakukan perhitungan untuk menentukan jumlah *offspring* yang dihasilkan dengan cara mengalikan nilai *crossover rate* dengan *popsi*.

Baris	Kode Program
1	<code>int anakCrossover = (int) Math.round(popSize * cr);</code>
2	<code>Individu crossover1[] = new Individu[anakCrossover];</code>
3	<code>Individu crossover2[] = new Individu[anakCrossover];</code>
4	<code>for (int i = 0; i &lt; crossover1.length; i++) {</code>
5	<code>    int cutPoint = random.nextInt((Kromosom1.length -</code>
6	<code>        2) - 1 + 1) + 1;</code>
7	<code>    int indexIndiv1 = random.nextInt((populasi1.length</code>
8	<code>        - 1) - 0 + 1) + 0;</code>
9	<code>    int indexIndiv2 = random.nextInt((populasi1.length</code>
10	<code>        - 1) - 0 + 1) + 0;</code>
11	
12	<code>    while (indexIndiv1 == indexIndiv2) {</code>
13	<code>        indexIndiv2 =</code>
14	<code>        random.nextInt((populasi1.length - 1) - 0 + 1)</code>
15	<code>        + 0;</code>
16	<code>    }</code>

```

17      System.out.println(indexIndiv1 + "\t" +
18      indexIndiv2);
19      System.out.println(cutPoint);
20
21      crossover1[i] = crossover(populasi1[indexIndiv1],
22      populasi1[indexIndiv2], cutPoint);
23      }
24      for (int i = 0; i < crossover2.length; i++) {
25          int cutPoint =
26          random.nextInt((Kromosom2.length - 2) - 1 + 1)
27          + 1;
28          int indexIndiv1 =
29          random.nextInt((populasi2.length - 1) - 0 + 1)
30          + 0;
31          int indexIndiv2 =
32          random.nextInt((populasi2.length - 1) - 0 + 1)
33          + 0;
34
35          while (indexIndiv1 == indexIndiv2) {
36              indexIndiv2 =
37              random.nextInt((populasi2.length - 1) -
38              0 + 1) + 0;
39          }
40          System.out.println(indexIndiv1 + "\t" +
41          indexIndiv2);
42          System.out.println(cutPoint);
43
44          crossover2[i] =
45          crossover(populasi2[indexIndiv1],
46          populasi2[indexIndiv2], cutPoint);
47      }

```

**Kode Program 5.2 Implementasi Proses Crossover**

Berikut penjelasan Kode Program 5.2:

1. Baris 1, Menghitung banyaknya *offspring* untuk *crossover*.
2. Baris 2-3, Inisialisasi array *crossover* sebanyak jumlah *offspring*.
3. Baris 4, Perulangan untuk menentukan untuk menentukan *cutPoint*, individu yang dipakai, dan proses *crossover*.
4. Baris 5-6, Inisialisasi nilai variabel *cut point* dengan nilai *random* dalam *range* 0 sampai panjang kromosom1 – 2
5. Baris 7-8, Inisialisasi nilai variabel *indexIndiv1* untuk menentukan individu pertama yang dipilih secara *random*.
6. Baris 9-10, Inisialisasi nilai variabel *indexIndiv2* untuk menentukan individu pertama yang dipilih secara *random*.
7. Baris 12-16, Perulangan untuk memilih ulang *indexIndiv2* jika *indexIndiv1* == *indexIndiv2*.

8. Baris 17-22, Menentukan nilai *crossover2* ke *i* dengan memanggil *method crossover* dengan parameter populasi2 ke - *indexIndiv1*, populasi2 ke - *indexIndiv2*, dan *cut point*.
9. Baris 24, Perulangan untuk menentukan untuk menentukan *cutPoint*, individu yang dipakai, dan proses *crossover*.
10. Baris 25-27, Inisialisasi nilai variabel *cut point* dengan nilai *random* dalam range 0 sampai panjang kromosom2 – 2.
11. Baris 28-30, Inisialisasi nilai variabel *indexIndiv1* untuk menentukan individu pertama yang dipilih secara *random*.
12. Baris 31-33, Inisialisasi nilai variabel *indexIndiv2* untuk menentukan individu pertama yang dipilih secara *random*.
13. Baris 35-39, Perulangan untuk memilih ulang *indexIndiv2* jika *indexIndiv1* == *indexIndiv2*.
14. Baris 40-47, Menentukan nilai *crossover2* ke *i* dengan memanggil *method crossover* dengan parameter populasi2 ke - *indexIndiv1*, populasi2 ke - *indexIndiv2*, dan *cut point*.

Baris	Kode Program
1	<code>public static Individu crossover(Individu ind1,</code>
2	<code>Individu ind2, int cutPoint) {</code>
3	<code>int[] indiv1 = ind1.getKromosom();</code>
4	<code>int[] indiv2 = ind2.getKromosom();</code>
5	<code>int[] child = new int[indiv1.length];</code>
6	
7	<code>for (int i = 0; i &lt; child.length; i++) {</code>
8	<code>if (i &lt; cutPoint) {</code>
9	<code>child[i] = indiv1[i];</code>
10	<code>} else {</code>
11	<code>for (int j = 0; j &lt; indiv2.length; j++) {</code>
12	<code>int y = indiv2[j];</code>
13	<code>boolean contains =</code>
14	<code>IntStream.of(child).anyMatch(x -&gt; x == y);</code>
15	<code>if (!contains) {</code>
16	<code>child[i++] = indiv2[j];</code>
17	<code>}</code>
18	<code>}</code>
19	<code>}</code>
20	<code>}</code>
21	
22	<code>Individu hasil = new Individu();</code>
23	<code>hasil.setKromosom(child);</code>
24	
25	<code>return hasil;</code>
26	<code>}</code>

**Kode Program 5.3 Implementasi Program Method Crossover**



Berikut penjelasan Kode Program 5.3:

1. Baris 1-2 , Membuat *method* yang bernama *crossover* dengan parameter *ind1*, *ind2*, *cutPoint*.
2. Baris 3-4, Inisialisasi variabel *indiv1* dengan nilai *ind1.getKromosom* dan inisialisasi variabel *indiv2* dengan nilai *ind2.getKromosom*.
3. Baris 5, Inisialisasi variabel *child* sebanyak *indiv1.length*.
4. Baris 7, Perulangan untuk melakukan *crossover*.
5. Baris 8-9, Seleksi jika *i* kurang dari *cutPoint* maka akan di-set nilai *child* ke *i* dengan nilai *indiv1* *i*.
6. Baris 10, Seleksi jika *i* lebih dari sama dengan *cutPoint*.
7. Baris 11-20, Perulangan untuk memasukkan nilai *Indiv2* ke *j* jika belum ada di *child* ke *i*.
8. Baris 22-23, Inisialisasi variabel hasil dan memanggil *method* *setKromosom* dengan parameter *child* pada variabel hasil.
9. Baris 25-26 , Mengembalikan nilai hasil.

### 5.2.3 Implementasi Proses Mutasi

Selanjutnya, proses yang kedua merupakan proses mutasi. Proses ini dilakukan dengan cara mengambil *parent* dari hasil inisialisasi, kemudian menentukan titik tukar pada masing-masing *parent* untuk melakukan pertukaran gen. Selain itu juga dilakukan perhitungan untuk menentukan jumlah *offspring* yang dihasilkan dengan cara mengalikan *mr* dengan *popsi*.

Baris	Kode Program
1	<code>int anakMutasi = (int) Math.round(popSize * mr);</code>
2	<code>Individu mutasi1[] = new Individu[anakMutasi];</code>
3	<code>Individu mutasi2[] = new Individu[anakMutasi];</code>
4	<code>for (int i = 0; i &lt; mutasi1.length; i++) {</code>
5	<code>    int indexIndiv = random.nextInt((populasi1.length</code>
6	<code>    - 1) - 0 + 1) + 0;</code>
7	<code>    mutasi1[i] = mutasi(populasi1[indexIndiv]);</code>
8	<code>}</code>
9	
10	<code>for (int i = 0; i &lt; mutasi2.length; i++) {</code>
11	<code>    int indexIndiv = random.nextInt((populasi2.length</code>
12	<code>    - 1) - 0 + 1) + 0;</code>
13	<code>    mutasi2[i] = mutasi(populasi2[indexIndiv]);</code>
14	<code>}</code>

**Kode Program 5.4 Implementasi Proses Mutasi**

Berikut penjelasan kode program 5.4:

1. Baris 1, Menghitung banyaknya *offspring* mutasi.
2. Baris 2-3, Inisialisasi variabel *mutasi1* dan variabel *mutasi2* sebanyak jumlah *offspring*.

3. Baris 5-8, Perulangan untuk melakukan mutasi untuk mutasi1 pada individu yang sudah dipilih secara *random*.
4. Baris 10-14, Perulangan untuk melakukan mutasi untuk mutasi2 pada individu yang sudah dipilih secara *random*.

Baris	Kode Program
1	<code>public static Individu mutasi(Individu indiv) {</code>
2	<code>    Random random = new Random();</code>
3	<code>    int ExchangePoint1 =</code>
4	<code>    random.nextInt((indiv.getKromosom().length - 1) - 0</code>
5	<code>    + 1) + 0;</code>
6	<code>    int ExchangePoint2 =</code>
7	<code>    random.nextInt((indiv.getKromosom().length - 1) - 0</code>
8	<code>    + 1) + 0;</code>
9	<code>    while (ExchangePoint1 == ExchangePoint2) {</code>
10	<code>        ExchangePoint2 =</code>
11	<code>        random.nextInt((indiv.getKromosom().length -</code>
12	<code>        1) - 0 + 1) + 0;</code>
13	<code>    }</code>
14	
15	<code>    int temp[] = new int[indiv.getKromosom().length];</code>
16	<code>    for (int i = 0; i &lt; indiv.getKromosom().length; i++)</code>
17	<code>    {</code>
18	<code>        temp[i] = indiv.getKromosom()[i];</code>
19	<code>    }</code>
20	<code>        int tempExchange = temp[ExchangePoint1];</code>
21	<code>        temp[ExchangePoint1] = temp[ExchangePoint2];</code>
22	<code>        temp[ExchangePoint2] = tempExchange;</code>
23	
24	<code>        Individu hasil = new Individu();</code>
25	<code>        hasil.setKromosom(temp);</code>
26	
27	<code>        return hasil;</code>
28	<code>    }</code>

**Kode Program 5.5 Implementasi Proses *Method* Mutasi**

Berikut penjelasan kode program 5.5:

1. Baris 1, Membuat *method* yang bernama mutasi dengan parameter Individu.
2. Baris 2-8, Inisialisasi dua buah *ExchangePoint* secara *random*.
3. Baris 9-13, Perulangan untuk memilih ulang *ExchangePoint* ke-2 jika masih sama dengan *ExchangePoint1*.
4. Baris 15-19, Inisialisasi variabel temp untuk menyimpan nilai dari kromosom *indiv*.
5. Baris 20-23, Proses melakukan penukaran kedua *ExchangePoint*.
6. Baris 24-25, Inisialisasi variabel hasil dan memanggil method setKromosom dengan parameter *temp*.
7. Baris 29 : Mengembalikan nilai hasil.

### 5.2.4 Implementasi Proses Evaluasi

Evaluasi merupakan proses ketiga dari algoritme genetika. Proses ini dilakukan dengan cara menggabungkan nilai *fitness* seluruh populasi hasil dari proses sebelumnya.

Baris	Kode Program
1	Individu mix1[] = new Individu[populasi1.length +
2	crossover1.length + mutasi1.length];
3	int index = 0;
4	for (Individu a : populasi1) {
5	mix1[index++] = a;
6	}
7	for (Individu a : crossover1) {
8	mix1[index++] = a;
9	}
10	for (Individu a : mutasi1) {
11	mix1[index++] = a;
12	}
13	System.out.println();
14	System.out.println("Hasil Mix");
15	for (Individu a : mix1) {
16	System.out.println(a + "\t" +
17	a.getFitness(jarakPagi));
18	}
19	Individu evaluasi1[] = new Individu[mix1.length];
20	for (int i = 0; i < mix1.length; i++) {
21	evaluasi1[i] = mix1[i];
22	}
23	Individu mix2[] = new Individu[populasi2.length +
24	crossover2.length + mutasi2.length];
25	index = 0;
26	for (Individu a : populasi2) {
27	mix2[index++] = a;
28	}
29	for (Individu a : crossover2) {
30	mix2[index++] = a;
31	}
32	for (Individu a : mutasi2) {
33	mix2[index++] = a;
34	}
35	System.out.println();
36	System.out.println("Hasil Mix");
37	for (Individu a : mix2) {
38	System.out.println(a + "\t" +
39	a.getFitness(jarakSiang));
40	}
41	Individu evaluasi2[] = new Individu[mix2.length];
42	for (int i = 0; i < mix2.length; i++) {
43	evaluasi2[i] = mix2[i];
44	}

Kode Program 5.6 Implementasi Proses Evaluasi

Berikut penjelasan Kode Program 5.6:

1. Baris 1, Inisialisasi variabel `mix1` sebanyak jumlah populasi1 ditambah jumlah *offspring crossover1*, *offspring mutasi1*.
2. Baris 3-18, Menentukan nilai `mix1` dengan menggabungkan seluruh nilai dari populasi1, *offspring crossover1*, dan *offspring mutasi1*.
3. Baris 19-22, Inisialisasi variabel `evaluasi1` sebanyak panjang `mix1` dan menentukan nilai `evaluasi1` dengan nilai `mix1`.
4. Baris 23-34, Inisialisasi variabel `mix2` sebanyak jumlah populasi2 ditambah jumlah *offspring crossover2*, *offspring mutasi2*.
5. Baris 35-40, Menentukan nilai `mix2` dengan menggabungkan seluruh nilai dari populasi2, *offspring crossover2*, dan *offspring mutasi2*.
6. Baris 41-44, Inisialisasi variabel `evaluasi2` sebanyak panjang `mix2` dan menentukan nilai `evaluasi2` dengan nilai `mix2`.

### 5.2.5 Implementasi Proses Seleksi

Selanjutnya tahapan terakhir merupakan proses seleksi. Proses ini dilakukan dengan cara mengambil nilai *fitness* yang sudah digabungkan dari proses evaluasi, kemudian melakukan *sorting* untuk mendapatkan nilai individu terbaik berdasarkan jumlah *popsiz*e dan nilai *fitness* tertinggi. Nilai yang sudah di-*sorting* akan digunakan sebagai *parent* di generasi selanjutnya.

Baris	Kode Program
1	<code>for (int i = 0; i &lt; evaluasi1.length - 1; i++) {</code>
2	<code>    for (int j = i; j &lt; evaluasi1.length; j++) {</code>
3	<code>        if (evaluasi1[i].getFitness(jarakPagi) &lt;</code>
4	<code>            evaluasi1[j].getFitness(jarakPagi)) {</code>
5	<code>            Individu temp = evaluasi1[i];</code>
6	<code>            evaluasi1[i] = evaluasi1[j];</code>
7	<code>            evaluasi1[j] = temp;</code>
8	<code>        }</code>
9	<code>    }</code>
10	<code>}</code>
11	<code>int[][] seleksil = new int[populasi1.length][];</code>
12	<code>for (int i = 0; i &lt; populasi1.length; i++) {</code>
13	<code>    seleksil[i] = evaluasi1[i].getKromosom();</code>
14	<code>}</code>
15	<code>for (int i = 0; i &lt; populasi1.length; i++) {</code>
16	<code>    populasi1[i].setKromosom(seleksil[i]);</code>
17	<code>}</code>
18	<code>for (int i = 0; i &lt; evaluasi2.length - 1; i++) {</code>
19	<code>    for (int j = i; j &lt; evaluasi2.length; j++) {</code>
20	<code>        if (evaluasi2[i].getFitness(jarakSiang) &lt;</code>
21	<code>            evaluasi2[j].getFitness(jarakSiang)) {</code>
22	<code>            Individu temp = evaluasi2[i];</code>
23	<code>            evaluasi2[i] = evaluasi2[j];</code>
24	<code>            evaluasi2[j] = temp;</code>
25	<code>        }</code>

```

26     }
27 }
28 int[][] seleksi2 = new int[populasi2.length][];
29 for (int i = 0; i < populasi2.length; i++) {
30     seleksi2[i] = evaluasi2[i].getKromosom();
31 }
32 for (int i = 0; i < populasi2.length; i++) {
33     populasi2[i].setKromosom(seleksi2[i]);
34 }

```

#### Kode Program 5.7 Implementasi Proses Seleksi

Berikut penjelasan Kode Program 5.7:

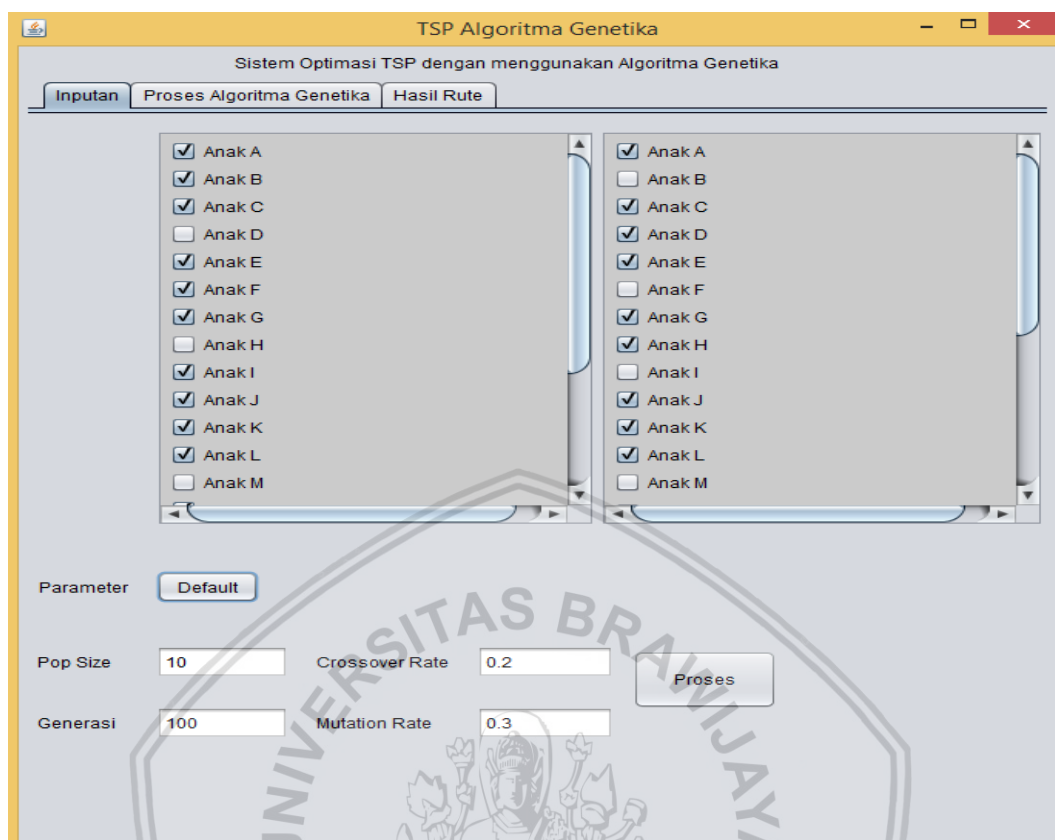
1. Baris 1-10, Perulangan untuk melakukan sorting variabel evaluasi dengan algoritme *bubbleSort*.
2. Baris 11, Inisialisasi variabel *seleksi1* sebanyak panjang populasi1 dan menentukan nilai *seleksi1* dengan nilai *evaluasi1.getKromosom*.
3. Baris 12-17, Perulangan untuk mengupdate populasi1 dengan nilai evaluasi1 dari *index* ke 0 sampai panjang populasi1-1.
4. Baris 18-27, Perulangan untuk melakukan *sorting* variabel evaluasi2 dengan algoritme *bubbleSort*.
5. Baris 28, Inisialisasi variabel *seleksi2* sebanyak panjang populasi2 dan menentukan nilai seleksi2 dengan nilai *evaluasi2.getKromosom*.
6. Baris 29-34, Perulangan untuk memutakhirkan populasi2 dengan nilai evaluasi2 dari *index* ke 0 sampai panjang populasi2-1.

### 5.3 Implementasi *User Interface*

Pada sistem ini, implementasi antarmuka terbagi menjadi beberapa halaman yaitu halaman inputan, halaman proses algoritme genetika, dan halaman hasil rute.

#### 5.3.1 Implementasi *User Interface* Halaman Utama

Pada implementasi *user interface* halaman utama terbagi menjadi 3 sub halaman yaitu halaman inputan, halaman proses algoritme genetika, dan halaman hasil rute. Pada halaman inputan terdapat beberapa fitur yaitu fitur pilih kloter, fitur centang siswa, *button default* parameter, *field input* nilai, *popsi*, generasi, *crossover rate* dan *mutation rate*, serta *button* proses untuk memulai proses algoritme genetika. *User Interface* Halaman Utama dapat dilihat pada Gambar 5.1.



Gambar 5.1 *User Interface* Halaman Utama

### 5.3.2 Implementasi *User Interface* Halaman Proses Algoritme Genetika

#### 5.3.2.1 Implementasi *User Interface* Halaman Populasi Awal

Pada implementasi *user interface* halaman populasi awal menampilkan data kromosom dari setiap generasi. Data populasi awal terdiri dari individu, kromosom, total jarak dan *fitness*. *User Interface* Halaman Populasi Awal dapat dilihat pada Gambar 5.2 dan Gambar 5.3.



Generasi ke-	Individu	Kromosom	Total Jarak	Fitness
1	P1	14 9 3 2 11 5 7 12 ...	28.2	3.546099290780142
	P2	1 2 7 3 5 11 12 14 ...	22.8	4.385964912280701
	P3	1 7 11 10 2 5 14 6 ...	27.09	3.6913990402362...
	P4	9 2 1 3 6 14 12 11 ...	20.73	4.82392667631452
	P5	19 3 12 5 7 14 6 2 ...	28.7500000000000...	3.478260869565217
	P6	10 14 3 1 7 2 11 5 ...	25.1	3.98406374501992
	P7	9 1 11 14 2 6 5 12 ...	26.9000000000000...	3.7174721189591...
	P8	3 1 14 2 7 9 10 6 1 ...	30.0	3.33333333333333...
	P9	11 2 1 14 7 10 3 12 ...	30.2	3.3112582781456...
	P10	14 2 11 9 5 6 3 10 ...	24.7599999999999...	4.0387722132471...
2	P1	12 9 3 1 5 7 14 6 2 ...	20.35	4.914004914004914
	P2	9 2 1 3 6 14 12 11 ...	20.73	4.82392667631452
	P3	6 2 7 3 5 11 12 14 ...	22.65	4.415011037527594
	P4	1 2 7 3 5 11 12 14 ...	22.8	4.385964912280701
	P5	14 2 11 9 5 6 3 10 ...	24.7599999999999...	4.0387722132471...
	P6	10 14 3 1 7 2 11 5 ...	25.1	3.98406374501992
	P7	11 2 1 7 10 5 14 6 ...	26.2000000000000...	3.8167938931297...
	P8	14 9 2 11 5 6 3 10 ...	26.2600000000000...	3.808073115003807
	P9	9 1 11 14 2 6 5 12 ...	26.9000000000000...	3.7174721189591...
	P10	1 7 11 10 2 5 14 6 ...	27.09	3.6913990402362...
3	P1	12 9 3 1 5 7 14 6 2 ...	20.35	4.914004914004914
	P2	9 2 1 3 6 14 12 11 ...	20.73	4.82392667631452
	P3	12 9 3 14 5 7 1 6 2 ...	21.65	4.618937644341802
	P4	6 2 7 3 5 11 12 14 ...	22.65	4.415011037527594
	P5	1 2 7 3 5 11 12 14 ...	22.8	4.385964912280701
	P6	1 2 7 3 5 11 12 14 ...	22.8	4.385964912280701
	P7	14 2 11 9 5 6 3 10 ...	24.7599999999999...	4.0387722132471...
	P8	10 14 3 1 7 2 11 5 ...	25.1	3.98406374501992
	P9	1 7 9 10 2 5 14 6 1 ...	25.49	3.9231071008238...
	P10	1 7 11 10 2 14 9 5 ...	25.6600000000000...	3.8971161340607...

Gambar 5.2 User Interface Populasi Awal Kloter Pagi

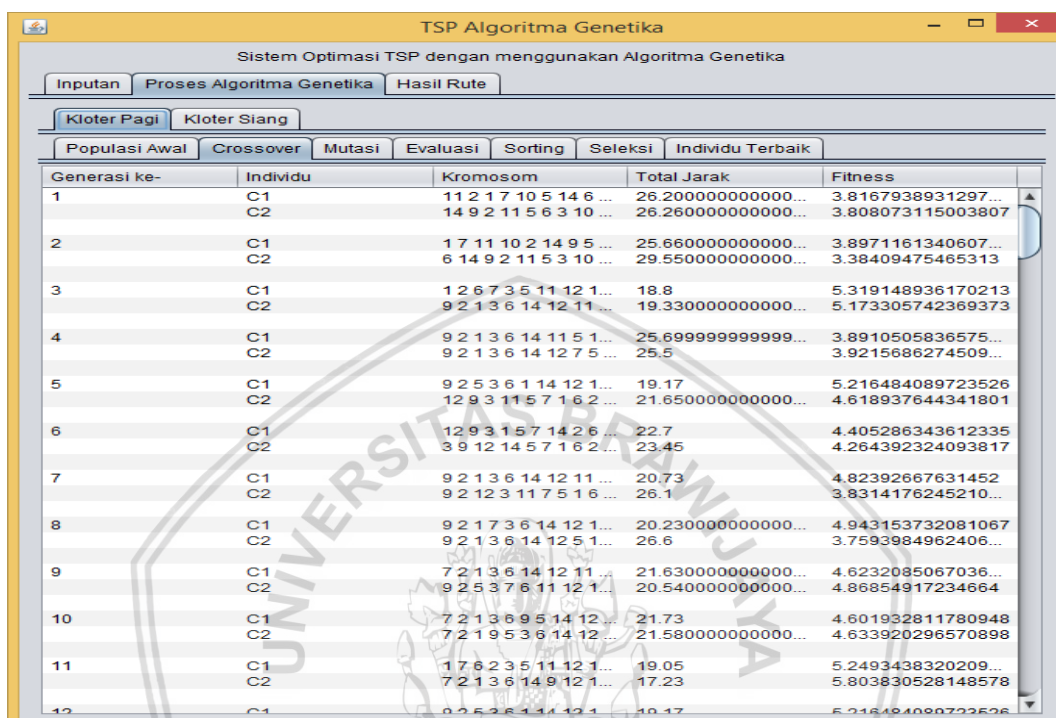
Generasi ke-	Individu	Kromosom	Total Jarak	Fitness
1	P1	11 1 7 8 4 3 5 12 10	28.0800000000000...	3.561253561253561
	P2	3 10 1 11 4 12 7 8 5	26.3800000000000...	3.790750568612585
	P3	10 12 4 1 7 8 5 3 11	28.6799999999999...	3.4867503486750...
	P4	8 10 4 11 12 7 1 3 5	23.4399999999999...	4.266211604095564
	P5	7 1 11 3 4 5 8 10 12	20.41	4.899559039686428
	P6	12 4 1 8 10 11 5 3 7	27.9599999999999...	3.5765379113018...
	P7	10 5 7 8 11 1 4 3 12	28.53	3.5050823694356...
	P8	5 7 4 11 3 10 12 8 1	27.0000000000000...	3.7037037037037...
	P9	10 4 3 7 8 12 11 1 5	25.1099999999999...	3.9824771007566...
	P10	12 10 11 5 7 3 4 1 8	28.45	3.51493848857645
2	P1	7 1 11 3 4 5 8 10 12	20.41	4.899559039686428
	P2	8 10 4 11 12 7 1 3 5	23.4399999999999...	4.266211604095564
	P3	10 4 3 7 8 12 11 1 5	25.1099999999999...	3.9824771007566...
	P4	12 4 1 3 10 11 5 8 7	25.93	3.8565368299267...
	P5	3 10 1 11 4 12 7 8 5	26.3800000000000...	3.790750568612585
	P6	5 7 4 11 3 10 12 8 1	27.0000000000000...	3.7037037037037...
	P7	5 7 4 11 3 10 12 8 1	27.0000000000000...	3.7037037037037...
	P8	12 4 1 8 10 11 5 3 7	27.9599999999999...	3.5765379113018...
	P9	11 1 7 8 4 3 5 12 10	28.0800000000000...	3.561253561253561
	P10	12 10 11 5 7 3 4 1 8	28.45	3.51493848857645
3	P1	12 4 5 8 10 11 1 3 7	17.36	5.760368663594471
	P2	3 10 1 11 4 5 7 8 12	19.6800000000000...	5.081300813008129
	P3	7 1 11 3 4 5 8 10 12	20.41	4.899559039686428
	P4	10 7 1 11 3 4 5 8 12	20.4300000000000...	4.894762604013705
	P5	8 10 4 11 12 7 1 3 5	23.4399999999999...	4.266211604095564
	P6	12 8 10 4 11 7 1 3 5	23.9099999999999...	4.182350480970306
	P7	5 7 4 11 3 1 12 8 10	24.8100000000000...	4.030632809351067
	P8	10 4 3 7 8 12 11 1 5	25.1099999999999...	3.9824771007566...
	P9	12 4 1 3 10 11 5 8 7	25.93	3.8565368299267...
	P10	3 10 1 11 4 12 7 8 5	26.3800000000000...	3.790750568612585

Gambar 5.3 User Interface Populasi Awal Kloter Siang



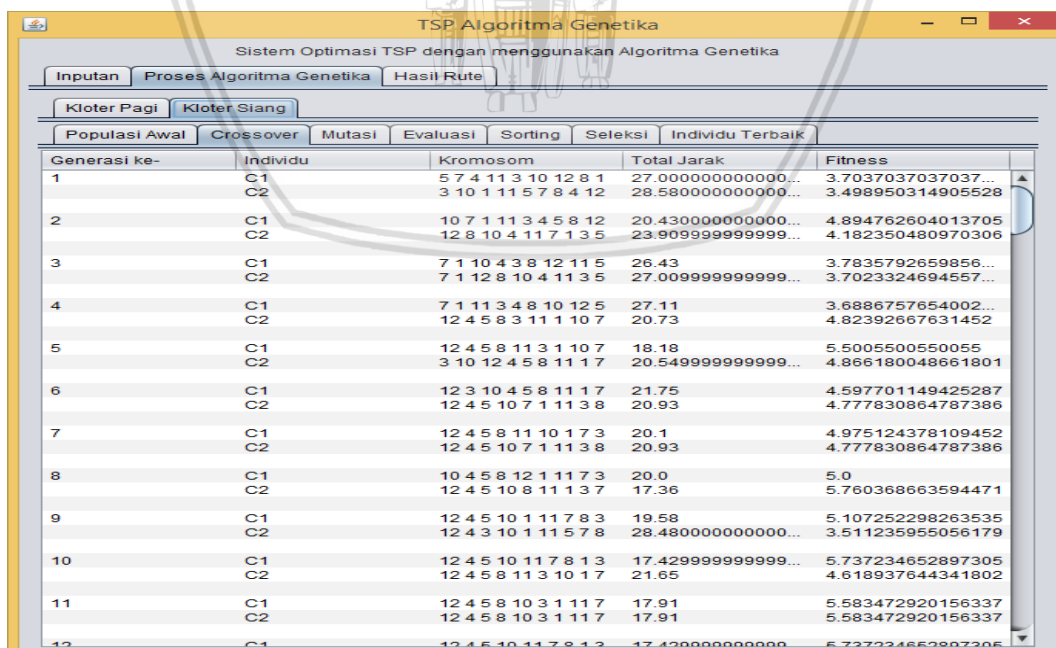
### 5.3.2.2 Implementasi User Interface Halaman Crossover

Pada implementasi *user interface* halaman *crossover* menampilkan data populasi hasil dari proses *crossover* dari masing-masing generasi. Data populasi awal terdiri dari individu, kromosom, total jarak dan *fitness*. *User Interface* Halaman *Crossover* dapat dilihat pada Gambar 5.4 dan Gambar 5.5.



Generasi ke-	Individu	Kromosom	Total Jarak	Fitness
1	C1	11 2 1 7 10 5 14 6 ...	26.2000000000000...	3.8167938931297...
	C2	14 9 2 11 5 6 3 10 ...	26.2600000000000...	3.808073115003807
2	C1	1 7 11 10 2 14 9 5 ...	25.6600000000000...	3.8971161340607...
	C2	6 14 9 2 11 5 3 10 ...	29.5500000000000...	3.38409475465313
3	C1	1 2 6 7 3 5 11 12 1 ...	18.8	5.319148936170213
	C2	9 2 1 3 6 14 12 11 ...	19.3300000000000...	5.173305742369373
4	C1	9 2 1 3 6 14 11 5 1 ...	25.6999999999999...	3.8910505836575...
	C2	9 2 1 3 6 14 12 7 5 ...	25.5	3.9215686274509...
5	C1	9 2 5 3 6 1 14 12 1 ...	19.17	5.216484089723526
	C2	12 9 3 11 5 7 1 6 2 ...	21.6500000000000...	4.618937644341801
6	C1	12 9 3 1 5 7 14 2 6 ...	22.7	4.405286343612335
	C2	3 9 12 14 5 7 1 6 2 ...	23.45	4.264392324093817
7	C1	9 2 1 3 6 14 12 11 ...	20.73	4.82392667631452
	C2	9 2 12 3 11 7 5 1 6 ...	26.1	3.8314176245210...
8	C1	9 2 1 7 3 6 14 12 1 ...	20.2300000000000...	4.943153732081067
	C2	9 2 1 3 6 14 12 5 1 ...	26.6	3.7593984962406...
9	C1	7 2 1 3 6 14 12 11 ...	21.6300000000000...	4.6232085067036...
	C2	9 2 5 3 7 6 11 12 1 ...	20.5400000000000...	4.86854917234664
10	C1	7 2 1 3 6 9 5 14 12 ...	21.73	4.601932811780948
	C2	7 2 1 9 5 3 6 14 12 ...	21.5800000000000...	4.633920296570898
11	C1	1 7 6 2 3 5 11 12 1 ...	19.05	5.2493438320209...
	C2	7 2 1 3 6 14 9 12 1 ...	17.23	5.803830528148578
12	C1	9 2 5 3 6 14 12 1 ...	19.17	5.216484089723526
	C2	...	...	...

Gambar 5.4 User Interface Crossover Kloter Pagi



Generasi ke-	Individu	Kromosom	Total Jarak	Fitness
1	C1	5 7 4 11 3 10 12 8 1	27.0000000000000...	3.7037037037037...
	C2	3 10 1 11 5 7 8 4 12	28.5800000000000...	3.498950314905528
2	C1	10 7 1 11 3 4 5 8 12	20.4300000000000...	4.894762604013705
	C2	12 8 10 4 11 7 1 3 5	23.9099999999999...	4.182350480970306
3	C1	7 1 10 4 3 8 12 11 5	26.43	3.7835792659856...
	C2	7 1 12 8 10 4 11 3 5	27.0099999999999...	3.7023324694557...
4	C1	7 1 11 3 4 8 10 12 5	27.11	3.6886757654002...
	C2	12 4 5 8 3 11 1 10 7	20.73	4.82392667631452
5	C1	12 4 5 8 11 3 1 10 7	18.18	5.5005500550055
	C2	3 10 12 4 5 8 11 1 7	20.5499999999999...	4.866180048661801
6	C1	12 3 10 4 5 8 11 1 7	21.75	4.597701149425287
	C2	12 4 5 10 7 1 11 3 8	20.93	4.777830864787386
7	C1	12 4 5 8 11 10 1 7 3	20.1	4.975124378109452
	C2	12 4 5 10 7 1 11 3 8	20.93	4.777830864787386
8	C1	10 4 5 8 12 1 11 7 3	20.0	5.0
	C2	12 4 5 10 8 11 1 3 7	17.36	5.760368663594471
9	C1	12 4 5 10 1 11 7 8 3	19.58	5.107252298263535
	C2	12 4 3 10 1 11 5 7 8	28.4800000000000...	3.511235955056179
10	C1	12 4 5 10 11 7 8 1 3	17.4299999999999...	5.737234652897305
	C2	12 4 5 8 11 3 10 1 7	21.65	4.618937644341802
11	C1	12 4 5 8 10 3 1 11 7	17.91	5.583472920156337
	C2	12 4 5 8 10 3 1 11 7	17.91	5.583472920156337
12	C1	12 4 5 10 11 7 8 1 3	17.4299999999999...	5.737234652897305
	C2	...	...	...

Gambar 5.5 User Interface Crossover Kloter Siang

### 5.3.2.3 Implementasi User Interface Halaman Mutasi

Pada implementasi *user interface* halaman mutasi menampilkan data populasi hasil dari proses mutasi dari masing-masing generasi. Data populasi awal terdiri dari individu, kromosom, total jarak dan *fitness*. *User Interface* Halaman Mutasi dapat dilihat pada Gambar 5.6 dan Gambar 5.7.

Generasi ke-	Individu	Kromosom	Total Jarak	Fitness
1	M1	12 9 3 1 5 7 14 6 2 ...	20.35	4.914004914004914
	M2	9 2 1 14 7 10 3 12 ...	30.4	3.2894736842105...
	M3	6 2 7 3 5 11 12 14 ...	22.65	4.415011037527594
2	M1	1 7 9 10 2 5 14 6 1 ...	25.49	3.9231071008238...
	M2	1 2 7 3 5 11 12 14 ...	22.8	4.385964912280701
	M3	12 9 3 14 5 7 1 6 2 ...	21.65	4.618937644341802
3	M1	6 2 7 3 5 11 14 12 ...	26.450000000000...	3.7807183364839...
	M2	1 7 9 10 2 5 6 14 1 ...	21.790000000000...	4.5892611289582...
	M3	14 2 11 9 5 6 1 10 ...	24.800000000000...	4.032258064516128
4	M1	12 9 3 11 5 7 1 6 2 ...	21.650000000000...	4.618937644341801
	M2	9 2 5 3 6 14 12 11 ...	21.370000000000...	4.679457182966775
	M3	3 9 12 14 5 7 1 6 2 ...	21.15	4.728132387706856
5	M1	12 9 3 11 7 5 1 6 2 ...	20.75	4.819277108433735
	M2	1 2 6 7 9 5 11 12 ...	22.4	4.464285714285714
	M3	9 2 5 11 6 14 12 3 ...	27.590000000000...	3.6245016310257...
6	M1	1 7 6 2 3 5 11 12 1 ...	18.45	5.420054200542006
	M2	12 9 1 11 5 7 3 6 2 ...	22.950000000000...	4.357298474945533
	M3	7 2 1 3 6 14 12 11 ...	17.73	5.640157924421883
7	M1	9 2 5 3 6 14 12 11 ...	21.470000000000...	4.657661853749417
	M2	2 9 5 3 6 1 14 12 1 ...	22.18	4.508566275924256
	M3	7 2 1 3 6 14 12 11 ...	19.63	5.094243504839532
8	M1	7 2 1 3 6 14 10 11 ...	22.3	4.484304932735426
	M2	9 2 5 3 6 1 14 7 11 ...	19.54	5.117707267144319
	M3	9 7 6 2 3 5 11 12 1 ...	19.45	5.141388174807198
9	M1	1 11 6 2 3 5 7 12 1 ...	26.05	3.838771593090211
	M2	10 2 1 2 6 14 12 11 ...	17.22	5.770210450085568
	M3			

Gambar 5.6 User Interface Mutasi Kloter Pagi

Generasi ke-	Individu	Kromosom	Total Jarak	Fitness
1	M1	11 1 7 4 8 3 5 12 10	29.2	3.4246575342465...
	M2	5 7 4 11 3 10 18 12	28.500000000000...	3.5087719298245...
	M3	12 4 1 3 10 11 5 8 7	25.93	3.8565368299267...
2	M1	3 10 1 11 4 5 7 8 12	19.680000000000...	5.081300813008129
	M2	12 4 5 8 10 11 1 3 7	17.36	5.760368663594471
	M3	5 7 4 11 3 1 12 8 10	24.810000000000...	4.030632809351067
3	M1	12 4 5 8 11 10 1 3 7	17.8	5.617977528089887
	M2	12 8 10 4 11 7 1 5 3	27.61	3.621876131836291
	M3	12 4 5 8 3 11 1 10 7	20.73	4.82392667631452
4	M1	10 5 1 11 3 4 7 8 12	28.48	3.5112359550561...
	M2	8 10 4 11 7 12 1 3 5	23.709999999999...	4.217629692113033
	M3	12 4 5 10 11 8 1 3 7	18.2	5.4945054945054...
5	M1	12 7 5 10 11 8 1 3 4	24.699999999999...	4.048582995951418
	M2	12 10 1 11 4 5 7 8 3	19.38	5.159958720330238
	M3	4 12 5 8 10 11 1 3 7	24.06	4.156275976724855
6	M1	12 4 5 8 3 11 1 10 7	20.73	4.82392667631452
	M2	12 10 1 11 4 5 7 8 3	19.38	5.159958720330238
	M3	3 10 4 11 1 5 7 8 12	28.080000000000...	3.561253561253561
7	M1	12 4 5 1 10 11 8 3 7	19.799999999999...	5.050505050505051
	M2	10 4 5 8 12 11 1 3 7	17.43	5.737234652897303
	M3	10 11 1 7 3 4 5 8 12	20.05	4.987531172069826
8	M1	4 12 5 1 10 11 8 3 7	26.499999999999...	3.773584905660378
	M2	12 10 1 11 4 5 8 7 3	19.18	5.213764337851929
	M3	12 10 5 11 4 1 7 8 3	27.78	3.599712023038157
9	M1	12 4 5 8 10 11 1 3 7	17.36	5.760368663594471
	M2	4 12 5 8 11 10 1 3 7	24.400000000000...	4.0816228520612...
	M3			

Gambar 5.7 User Interface Mutasi Kloter Siang

### 5.3.2.4 Implementasi User Interface Halaman Evaluasi

Pada implementasi *user interface* halaman evaluasi menampilkan penggabungan *parent* dan *child* yang dihasilkan pada ketiga proses sebelumnya. Data evaluasi terdiri dari individu, kromosom, total jarak dan *fitness*. *User Interface* Halaman evaluasi dapat dilihat pada Gambar 5.7 dan Gambar 5.8.

TSP Algoritma Genetika

Sistem Optimasi TSP dengan menggunakan Algoritma Genetika

Inputan

Proses Algoritma Genetika

Hasil Rute

Kloter Pagi

Kloter Siang

Populasi Awal

Crossover

Mutasi

Evaluasi

Sorting

Seleksi

Individu Terbaik

Generasi ke-	Individu	Kromosom	Total Jarak	Fitness
1	P1	14932115712...	28.2	3.546099290780142
	P2	12735111214...	22.8	4.385964912280701
	P3	17111025146...	27.09	3.6913990402362...
	P4	92136141211...	20.73	4.82392667631452
	P5	19312571462...	28.7500000000000...	3.478260869565217
	P6	10143172115...	25.1	3.98406374501992
	P7	91111426512...	26.9000000000000...	3.7174721189591...
	P8	31142791061...	30.0	3.33333333333333...
	P9	112114710312...	30.2	3.3112582781456...
	P10	14211956310...	24.7599999999999...	4.0387722132471...
2	C1	11217105146...	26.2000000000000...	3.8167938931297...
	C2	14921156310...	26.2600000000000...	3.808073115003807
	M1	12931571462...	20.35	4.914004914004914
	M2	92114710312...	30.4	3.2894736842105...
	M3	62735111214...	22.65	4.415011037527594
	P1	12931571462...	20.35	4.914004914004914
	P2	92136141211...	20.73	4.82392667631452
	P3	62735111214...	22.65	4.415011037527594
	P4	12735111214...	22.8	4.385964912280701
	P5	14211956310...	24.7599999999999...	4.0387722132471...
3	P1	12931571462...	20.35	4.914004914004914
	P2	92136141211...	20.73	4.82392667631452

Gambar 5.8 User Interface Evaluasi Kloter Pagi

Generasi ke-	Individu	Kromosom	Total Jarak	Fitness
1	P1	11 1 7 8 4 3 5 12 10 ...	28.0800000000000...	3.561253561253561
	P2	3 10 1 11 4 12 7 8 5 ...	26.3800000000000...	3.790750568612585
	P3	10 12 4 1 7 8 5 3 11 ...	28.6799999999999...	3.4867503486750...
	P4	8 10 4 1 11 12 7 1 3 5 ...	23.4399999999999...	4.266211604095564
	P5	7 1 11 3 4 5 8 10 12 ...	20.41	4.899559039686428
	P6	12 4 1 8 10 11 5 3 7 ...	27.9599999999999...	3.5765379113018...
	P7	10 5 7 8 11 1 4 3 12 ...	28.53	3.5050823694356...
	P8	5 7 4 11 3 10 12 8 1 ...	27.0000000000000...	3.7037037037037...
	P9	10 4 3 7 8 12 11 1 5 ...	25.1099999999999...	3.9824771007566...
	P10	12 10 11 5 7 3 4 1 8 ...	28.45	3.51493848857645
2	C1	5 7 4 11 3 10 12 8 1 ...	27.0000000000000...	3.7037037037037...
	C2	3 10 1 11 5 7 8 4 12 ...	28.5800000000000...	3.498950314905528
	M1	11 1 7 4 3 5 12 10 ...	29.2	3.4246575342465...
	M2	5 7 4 11 3 10 1 8 12 ...	28.5000000000000...	3.5087719298245...
	M3	12 4 1 3 10 11 5 8 7 ...	25.93	3.8565368299267...
	P1	7 1 11 3 4 5 8 10 12 ...	20.41	4.899559039686428
	P2	8 10 4 1 11 12 7 1 3 5 ...	23.4399999999999...	4.266211604095564
	P3	10 4 3 7 8 12 11 1 5 ...	25.1099999999999...	3.9824771007566...
	P4	12 4 1 3 10 11 5 8 7 ...	25.93	3.8565368299267...
	P5	3 10 1 11 4 12 7 8 5 ...	26.3800000000000...	3.790750568612585
3	P1	5 7 4 11 3 10 12 8 1 ...	27.0000000000000...	3.7037037037037...
	P2	5 7 4 11 3 10 12 8 1 ...	27.0000000000000...	3.7037037037037...
	P3	12 4 1 8 10 11 5 3 7 ...	27.9599999999999...	3.5765379113018...
	P4	11 1 7 8 4 3 5 12 10 ...	28.0800000000000...	3.561253561253561
	P5	12 10 11 5 7 3 4 1 8 ...	28.45	3.51493848857645
	P6	10 7 1 11 3 4 5 8 12 ...	20.4300000000000...	4.894762604013705
	P7	12 8 10 4 11 7 1 3 5 ...	23.9099999999999...	4.182350480970306
	P8	3 10 1 11 4 5 7 8 12 ...	19.6800000000000...	5.081300813008129
	P9	12 4 5 8 10 11 1 3 7 ...	17.36	5.760368663594471
	P10	5 7 4 11 3 1 12 8 10 ...	24.8100000000000...	4.030632809351067

Gambar 5.9 User Interface Evaluasi Kloter Siang

### 5.3.2.5 Implementasi User Interface Halaman Seleksi

Pada implementasi *user interface* halaman seleksi mengumpulkan populasi awal dan hasil *offspring* reproduksi ke dalam satu penampungan, kemudian mengurutkan nilai *fitness* mulai dari yang tertinggi hingga terendah sebanyak jumlah *popsiz*. Data populasi awal terdiri dari individu, kromosom, total jarak dan *fitness*. *User Interface* Halaman Populasi Awal dapat dilihat pada Gambar 5.10 dan Gambar 5.11.

Generasi ke-	Individu	Kromosom	Total Jarak	Fitness
1	P1	12 9 3 1 5 7 14 6 2 ...	20.35	4.914004914004914
	P2	9 2 1 3 6 14 12 11 ...	20.73	4.82392667631452
	P3	6 2 7 3 5 11 12 14 ...	22.65	4.415011037527594
	P4	1 2 7 3 5 11 12 14 ...	22.8	4.385964912280701
	P5	14 2 11 9 5 6 3 10 ...	24.759999999999999	4.0387722132471...
	P6	10 14 3 1 7 2 11 5 ...	25.1	3.98406374501992
	P7	11 2 1 7 10 5 14 6 ...	26.200000000000000	3.8167938931297...
	P8	14 9 2 11 5 6 3 10 ...	26.260000000000000	3.808073115003807
	P9	9 1 11 14 2 6 5 12 ...	26.900000000000000	3.7174721189591...
	P10	1 7 11 10 2 5 14 6 ...	27.09	3.6913990402362...
2	P1	12 9 3 1 5 7 14 6 2 ...	20.35	4.914004914004914
	P2	9 2 1 3 6 14 12 11 ...	20.73	4.82392667631452
	P3	12 9 3 14 5 7 1 6 2 ...	21.65	4.618937644341802
	P4	6 2 7 3 5 11 12 14 ...	22.65	4.415011037527594
	P5	1 2 7 3 5 11 12 14 ...	22.8	4.385964912280701
	P6	12 7 3 5 11 12 14 ...	22.8	4.385964912280701
	P7	14 2 11 9 5 6 3 10 ...	24.759999999999999	4.0387722132471...
	P8	10 14 3 1 7 2 11 5 ...	25.1	3.98406374501992
	P9	1 7 9 10 2 5 14 6 1 ...	25.49	3.9231071008238...
	P10	1 7 11 10 2 14 9 5 ...	25.660000000000000	3.8971161340607...
3	P1	1 2 6 7 3 5 11 12 1 ...	18.8	5.319148936170213
	P2	9 2 1 3 6 14 12 11 ...	19.330000000000000	5.173305742369373
	P3	12 9 3 1 5 7 14 6 2 ...	20.35	4.914004914004914
	P4	9 2 1 3 6 14 12 11 ...	20.73	4.82392667631452
	P5	12 9 3 14 5 7 1 6 2 ...	21.65	4.618937644341802
	P6	1 7 9 10 2 5 14 6 1 ...	21.790000000000000	4.5892611289582...
	P7	6 2 7 3 5 11 12 14 ...	22.65	4.415011037527594
	P8	1 2 7 3 5 11 12 14 ...	22.8	4.385964912280701
	P9	1 2 7 3 5 11 12 14 ...	22.8	4.385964912280701
	P10	14 2 11 9 5 6 3 10 ...	24.759999999999999	4.0387722132471...

Gambar 5.10 User Interface Seleksi Kloter Pagi

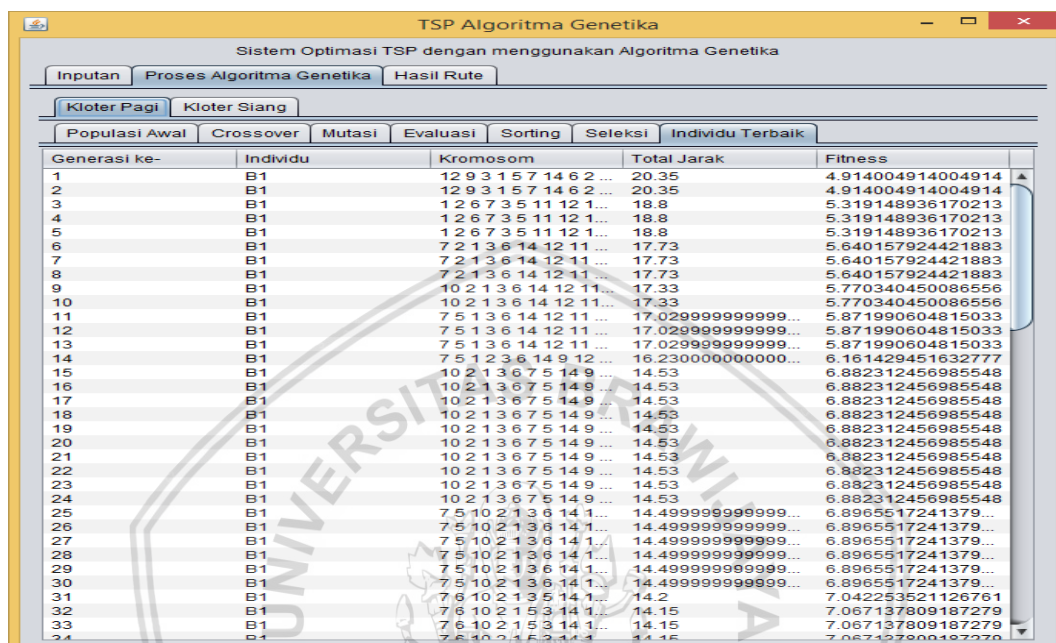
Generasi ke-	Individu	Kromosom	Total Jarak	Fitness
1	P1	7 1 11 3 4 5 8 10 12	20.41	4.899559039686428
	P2	8 10 4 11 12 7 1 3 5	23.439999999999999	4.266211604095564
	P3	10 4 3 7 8 12 11 1 5	25.109999999999999	3.9824771007566...
	P4	12 4 1 3 10 11 5 8 7	25.93	3.8565368299267...
	P5	3 10 1 11 4 12 7 8 5	26.380000000000000	3.790750568612585
	P6	5 7 4 11 3 10 12 8 1	27.000000000000000	3.7037037037037...
	P7	5 7 4 11 3 10 12 8 1	27.000000000000000	3.7037037037037...
	P8	12 4 1 8 10 11 5 3 7	27.959999999999999	3.5765379113018...
	P9	11 1 7 8 4 3 5 12 10	28.080000000000000	3.561253561253561
	P10	12 10 11 5 7 3 4 1 8	28.45	3.51493848857645
2	P1	12 4 5 8 10 11 1 3 7	17.36	5.760368863594471
	P2	3 10 1 11 4 5 7 8 12	19.680000000000000	5.081300813008129
	P3	7 1 11 3 4 5 8 10 12	20.41	4.899559039686428
	P4	10 7 1 11 3 4 5 8 12	20.430000000000000	4.894762604013705
	P5	8 10 4 11 12 7 1 3 5	23.439999999999999	4.266211604095564
	P6	12 8 10 4 11 7 1 3 5	23.909999999999999	4.182350480970306
	P7	5 7 4 11 3 1 12 8 10	24.810000000000000	4.030632809351067
	P8	10 4 3 7 8 12 11 1 5	25.109999999999999	3.9824771007566...
	P9	12 4 1 3 10 11 5 8 7	25.93	3.8565368299267...
	P10	3 10 1 11 4 12 7 8 5	26.380000000000000	3.790750568612585
3	P1	12 4 5 8 10 11 1 3 7	17.36	5.760368863594471
	P2	12 4 5 8 11 10 1 3 7	17.8	5.617977528089887
	P3	3 10 1 11 4 5 7 8 12	19.680000000000000	5.081300813008129
	P4	7 1 11 3 4 5 8 10 12	20.41	4.899559039686428
	P5	10 7 1 11 3 4 5 8 12	20.430000000000000	4.894762604013705
	P6	12 4 5 8 3 11 1 10 7	20.73	4.82392667631452
	P7	8 10 4 11 12 7 1 3 5	23.439999999999999	4.266211604095564
	P8	12 8 10 4 11 7 1 3 5	23.909999999999999	4.182350480970306
	P9	5 7 4 11 3 1 12 8 10	24.810000000000000	4.030632809351067
	P10	10 4 3 7 8 12 11 1 5	25.109999999999999	3.9824771007566...

Gambar 5.11 User Interface Seleksi Kloter Siang



### 5.3.2.6 Implementasi User Interface Halaman Individu Terbaik

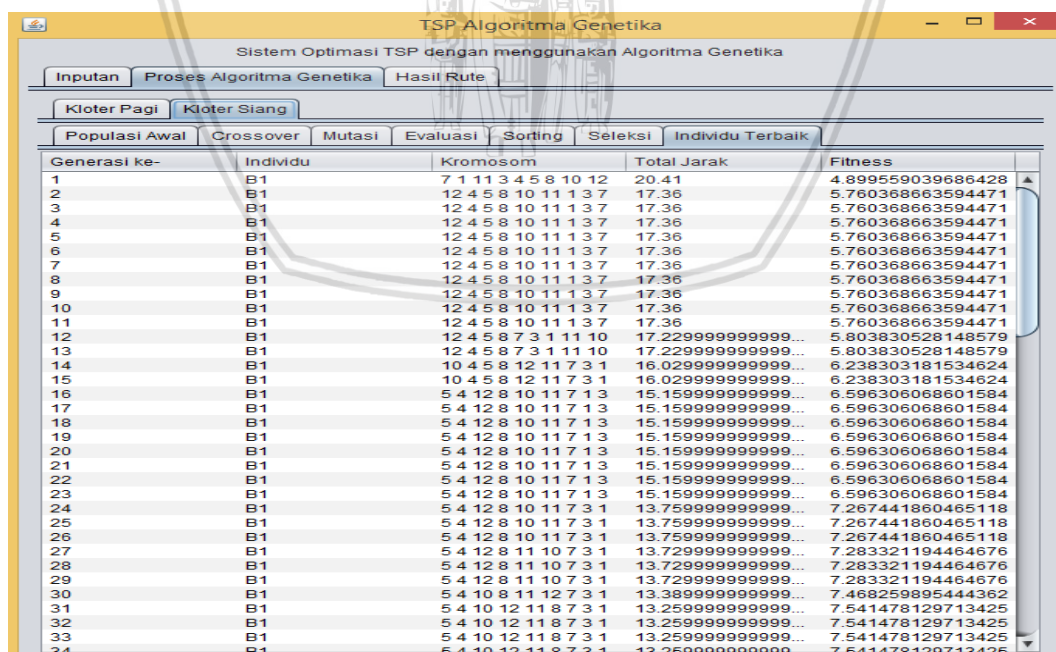
Pada implementasi *user interface* halaman individu terbaik menampilkan data individu-individu terbaik dari proses algoritme genetika dan menampilkan nilai *fitness*-nya. Data populasi awal terdiri dari individu, kromosom, total jarak dan *fitness*. *User Interface* Halaman Individu Terbaik dapat dilihat pada Gambar 5.12 dan Gambar 5.13.



The screenshot shows the 'TSP Algoritma Genetika' window with the 'Individu Terbaik' tab selected. The table displays the best individuals from the 'Kloter Pagi' cluster. The columns are: Generasi ke- (Generation), Individu (Individual), Kromosom (Chromosome), Total Jarak (Total Distance), and Fitness. The data shows a progression from generation 1 to 34, with fitness values increasing from approximately 4.91 to 7.06.

Generasi ke-	Individu	Kromosom	Total Jarak	Fitness
1	B1	12 9 3 1 5 7 14 6 2 ...	20.35	4.914004914004914
2	B1	12 9 3 1 5 7 14 6 2 ...	20.35	4.914004914004914
3	B1	1 2 6 7 3 5 11 12 1 ...	18.8	5.319148936170213
4	B1	1 2 6 7 3 5 11 12 1 ...	18.8	5.319148936170213
5	B1	1 2 6 7 3 5 11 12 1 ...	18.8	5.319148936170213
6	B1	7 2 1 3 6 14 12 11 ...	17.73	5.640157924421883
7	B1	7 2 1 3 6 14 12 11 ...	17.73	5.640157924421883
8	B1	7 2 1 3 6 14 12 11 ...	17.73	5.640157924421883
9	B1	10 2 1 3 6 14 12 11 ...	17.33	5.770340450086556
10	B1	10 2 1 3 6 14 12 11 ...	17.33	5.770340450086556
11	B1	7 5 1 3 6 14 12 11 ...	17.029999999999999	5.871990604815033
12	B1	7 5 1 3 6 14 12 11 ...	17.029999999999999	5.871990604815033
13	B1	7 5 1 3 6 14 12 11 ...	17.029999999999999	5.871990604815033
14	B1	7 5 1 2 3 6 14 9 12 ...	16.230000000000000	6.161429451632777
15	B1	10 2 1 3 6 7 5 14 9 ...	14.53	6.882312456985548
16	B1	10 2 1 3 6 7 5 14 9 ...	14.53	6.882312456985548
17	B1	10 2 1 3 6 7 5 14 9 ...	14.53	6.882312456985548
18	B1	10 2 1 3 6 7 5 14 9 ...	14.53	6.882312456985548
19	B1	10 2 1 3 6 7 5 14 9 ...	14.53	6.882312456985548
20	B1	10 2 1 3 6 7 5 14 9 ...	14.53	6.882312456985548
21	B1	10 2 1 3 6 7 5 14 9 ...	14.53	6.882312456985548
22	B1	10 2 1 3 6 7 5 14 9 ...	14.53	6.882312456985548
23	B1	10 2 1 3 6 7 5 14 9 ...	14.53	6.882312456985548
24	B1	10 2 1 3 6 7 5 14 9 ...	14.53	6.882312456985548
25	B1	7 5 10 2 1 3 6 14 1 ...	14.499999999999999	6.8965517241379...
26	B1	7 5 10 2 1 3 6 14 1 ...	14.499999999999999	6.8965517241379...
27	B1	7 5 10 2 1 3 6 14 1 ...	14.499999999999999	6.8965517241379...
28	B1	7 5 10 2 1 3 6 14 1 ...	14.499999999999999	6.8965517241379...
29	B1	7 5 10 2 1 3 6 14 1 ...	14.499999999999999	6.8965517241379...
30	B1	7 5 10 2 1 3 6 14 1 ...	14.499999999999999	6.8965517241379...
31	B1	7 6 10 2 1 5 3 14 1 ...	14.2	7.042253521126761
32	B1	7 6 10 2 1 5 3 14 1 ...	14.15	7.067137809187279
33	B1	7 6 10 2 1 5 3 14 1 ...	14.15	7.067137809187279
34	B1	7 6 10 2 1 5 3 14 1 ...	14.15	7.067137809187279

Gambar 5.13 User Interface Individu Terbaik Kloter Pagi



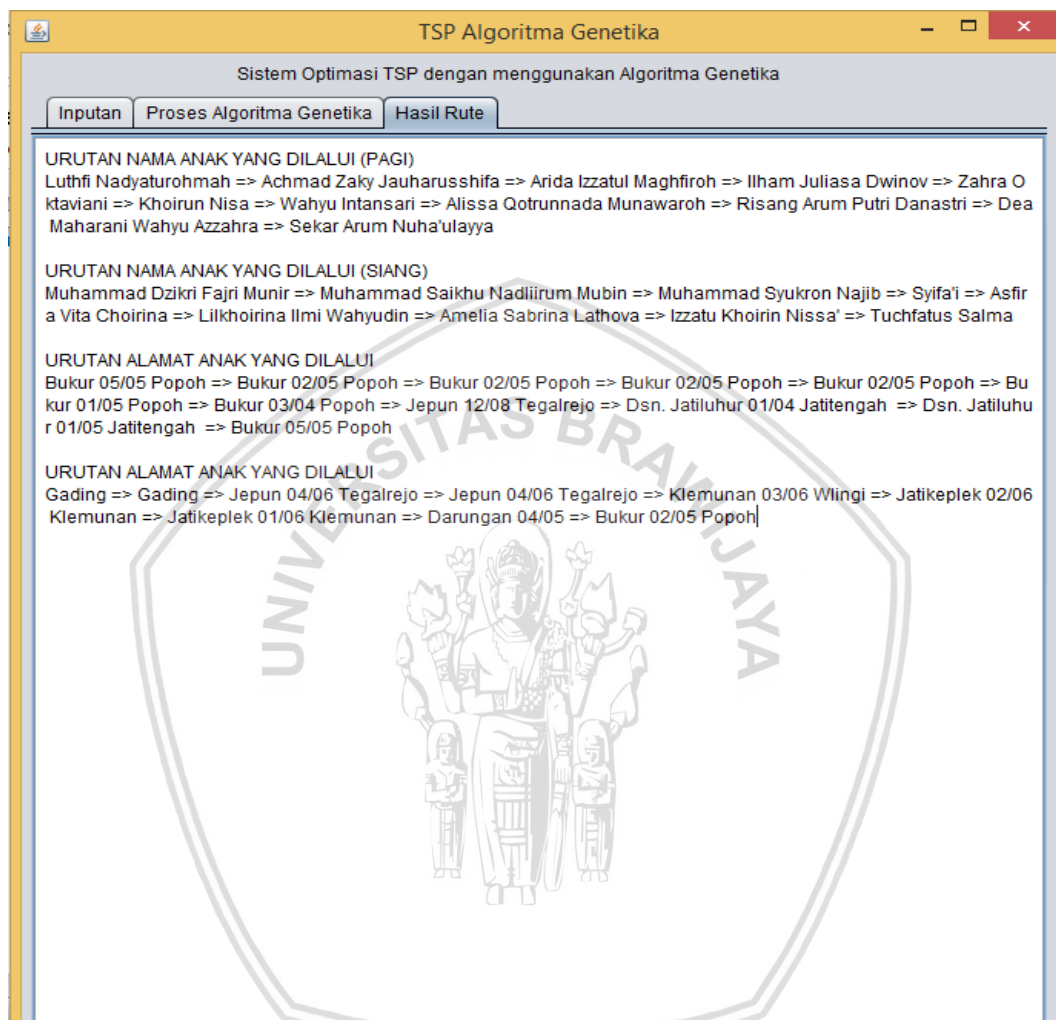
The screenshot shows the 'TSP Algoritma Genetika' window with the 'Individu Terbaik' tab selected. The table displays the best individuals from the 'Kloter Siang' cluster. The columns are: Generasi ke- (Generation), Individu (Individual), Kromosom (Chromosome), Total Jarak (Total Distance), and Fitness. The data shows a progression from generation 1 to 34, with fitness values increasing from approximately 4.89 to 7.54.

Generasi ke-	Individu	Kromosom	Total Jarak	Fitness
1	B1	7 1 11 3 4 5 8 10 12	20.41	4.899559039686428
2	B1	12 4 5 8 10 11 1 3 7	17.36	5.760368663594471
3	B1	12 4 5 8 10 11 1 3 7	17.36	5.760368663594471
4	B1	12 4 5 8 10 11 1 3 7	17.36	5.760368663594471
5	B1	12 4 5 8 10 11 1 3 7	17.36	5.760368663594471
6	B1	12 4 5 8 10 11 1 3 7	17.36	5.760368663594471
7	B1	12 4 5 8 10 11 1 3 7	17.36	5.760368663594471
8	B1	12 4 5 8 10 11 1 3 7	17.36	5.760368663594471
9	B1	12 4 5 8 10 11 1 3 7	17.36	5.760368663594471
10	B1	12 4 5 8 10 11 1 3 7	17.36	5.760368663594471
11	B1	12 4 5 8 10 11 1 3 7	17.36	5.760368663594471
12	B1	12 4 5 8 7 3 1 11 10	17.229999999999999	5.803830528148579
13	B1	12 4 5 8 7 3 1 11 10	17.229999999999999	5.803830528148579
14	B1	10 4 5 8 12 11 7 3 1	16.029999999999999	6.238303181534624
15	B1	10 4 5 8 12 11 7 3 1	16.029999999999999	6.238303181534624
16	B1	5 4 12 8 10 11 7 1 3	15.159999999999999	6.596306068601584
17	B1	5 4 12 8 10 11 7 1 3	15.159999999999999	6.596306068601584
18	B1	5 4 12 8 10 11 7 1 3	15.159999999999999	6.596306068601584
19	B1	5 4 12 8 10 11 7 1 3	15.159999999999999	6.596306068601584
20	B1	5 4 12 8 10 11 7 1 3	15.159999999999999	6.596306068601584
21	B1	5 4 12 8 10 11 7 1 3	15.159999999999999	6.596306068601584
22	B1	5 4 12 8 10 11 7 1 3	15.159999999999999	6.596306068601584
23	B1	5 4 12 8 10 11 7 1 3	15.159999999999999	6.596306068601584
24	B1	5 4 12 8 10 11 7 1 3	13.759999999999999	7.267441880465118
25	B1	5 4 12 8 10 11 7 1 3	13.759999999999999	7.267441880465118
26	B1	5 4 12 8 10 11 7 1 3	13.759999999999999	7.267441880465118
27	B1	5 4 12 8 11 10 7 3 1	13.729999999999999	7.283321194464676
28	B1	5 4 12 8 11 10 7 3 1	13.729999999999999	7.283321194464676
29	B1	5 4 12 8 11 10 7 3 1	13.729999999999999	7.283321194464676
30	B1	5 4 10 8 11 12 7 3 1	13.389999999999999	7.468259895444362
31	B1	5 4 10 12 11 8 7 3 1	13.259999999999999	7.541478129713425
32	B1	5 4 10 12 11 8 7 3 1	13.259999999999999	7.541478129713425
33	B1	5 4 10 12 11 8 7 3 1	13.259999999999999	7.541478129713425
34	B1	5 4 10 12 11 8 7 3 1	13.259999999999999	7.541478129713425

Gambar 5.14 User Interface Individu Terbaik Kloter Siang

### 5.3.2.7 Implementasi User Interface Halaman Hasil Rute

Pada implementasi *user interface* halaman hasil rute menampilkan hasil optimasi berupa rekomendasi rute yang berisi nama siswa dan alamat siswa. Data populasi awal terdiri dari individu, kromosom, total jarak dan *fitness*. *User Interface* Halaman Hasil Rute dapat dilihat pada Gambar 5.14.



Gambar 5.16 User Interface Hasil Rute Kloter Siang

## BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini menguraikan tentang pengujian yang akan dilakukan pada sistem. Hasil pengujian didapatkan berdasarkan skenario pengujian yang telah dibuat sebelumnya. Pengujian yang digunakan pada sistem ini terdiri dari pengujian ukuran populasi, pengujian kombinasi nilai *cr* dan *mr*, dan pengujian konvergen.

### 6.1 Pengujian Ukuran Populasi

Pengujian ini dilakukan menggunakan nilai dengan kelipatan 10 dimulai dari 10 hingga 100 sebanyak 10 kali uji coba. Jumlah masukkan data siswa yang digunakan pada pengujian ini sebanyak 10 data siswa kloter pagi dan 12 data siswa kloter siang. Hasil pengujian ukuran populasi kloter pagi dapat dilihat pada Tabel 6.1. dan hasil selengkapnya dapat dilihat pada Lampiran C.1.

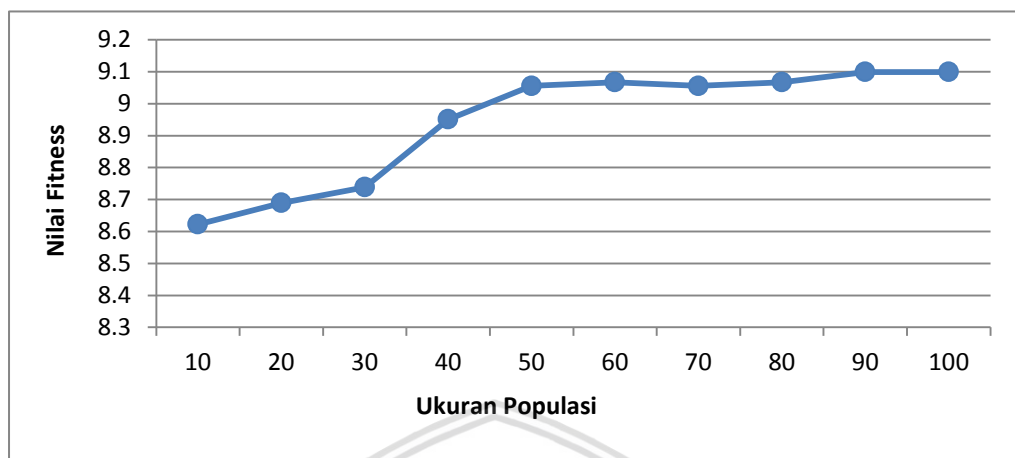
**Tabel 6.1 Hasil Pengujian Ukuran Populasi Kloter Pagi**

Populasi	Pengujian ke - <i>i</i>					Nilai Fitness
	1	2	....	9	10	
10	9,0991	9,0991	....	8,779	8,665	8,6225
20	8,779	9,0991	....	9,0991	8,665	8,6892
30	8,665	9,0991	....	9,0991	8,779	8,7384
40	8,779	9,0991	....	8,375	8,665	8,9512
50	9,0991	9,0991	....	9,0991	9,0991	9,0556
60	9,0991	9,0991	....	8,779	9,0991	9,0677
70	9,0991	9,0991	....	9,0991	9,0991	9,0556
80	9,0991	9,0991	....	9,0991	9,0991	9,0677
90	9,0991	9,0991	....	9,0991	9,0991	9,0991
100	9,0991	9,0991	....	9,0991	9,0991	9,0991

Dapat dilihat pada Tabel 6.1 bahwa hasil pengujian ukuran populasi menghasilkan rata-rata nilai *fitness* tertinggi sebesar 9,0991 yang mana terdapat pada ukuran populasi 100 dan rata-rata nilai *fitness* terendah yang dihasilkan adalah 8,6225 yaitu pada ukuran populasi 10. Hal ini disebabkan ukuran populasi yang semakin besar akan menghasilkan individu yang banyak. Semakin banyak individu maka semakin beragam variasi yang dihasilkan sehingga nilai *fitness* yang dihasilkan semakin besar dan semakin baik (Kusumaningsih, 2016).



Hasil pengujian ukuran populasi kloter pagi pada Tabel 6.1 maka grafik hasil pengujian ukuran populasi kloter pagi dapat dilihat pada Gambar 6.1.



**Gambar 6.1 Grafik Hasil Pengujian Ukuran Populasi Kloter Pagi**

Grafik Gambar 6.1 merupakan hasil pengujian ukuran populasi kloter pagi. Pengujian ukuran populasi dilakukan dengan cara menjumlahkan nilai *fitness* dari masing-masing populasi dimana nilai ukuran populasi merupakan nilai kelipatan yang dimulai dari kelipatan 10 sampai 100. Kemudian ditampilkan dalam bentuk grafik sehingga dapat diketahui nilai *fitness* tertinggi berada pada ukuran populasi ke-100 dengan nilai sebesar 9,0991.

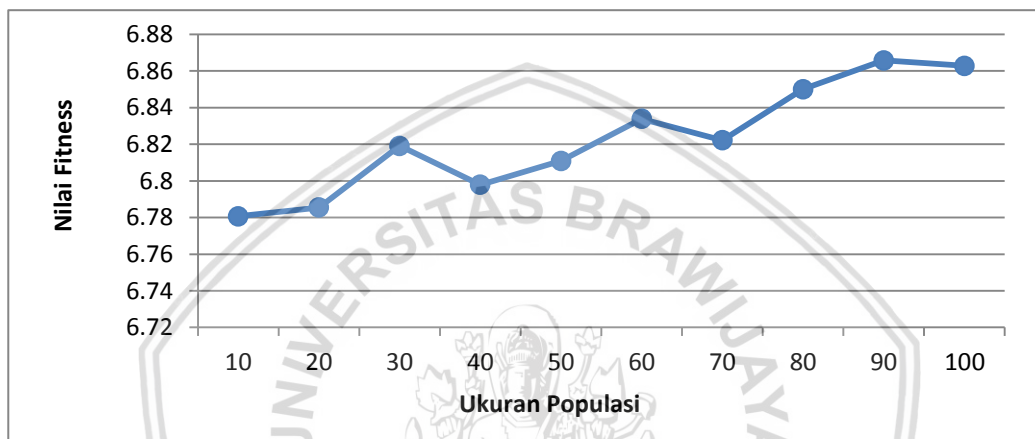
Selanjutnya hasil pengujian ukuran populasi pada kloter siang dapat dilihat pada Tabel 6.2 dan hasil selengkapnya dapat dilihat pada Lampiran C.1.

**Tabel 6.2 Hasil Pengujian Ukuran Populasi Kloter Siang**

Populasi	Pengujian ke - <i>i</i>					Nilai Fitness
	1	2	....	9	10	
10	6,7294	6,775	....	6,706	6,788	6,78064
20	6,706	6,882	....	6,821	6,706	6,7855
30	6,835	6,835	....	6,72	6,835	6,8191
40	6,882	6,882	....	6,835	6,752	6,7979
50	6,835	6,722	....	6,835	6,765	6,8109
60	6,835	6,835	....	6,835	6,765	6,8339
70	6,882	6,835	....	6,882	6,706	6,8222
80	6,765	6,868	....	6,882	6,882	6,8501
90	6,882	6,882	....	6,882	6,882	6,8658
100	6,882	6,882	....	6,882	6,821	6,8628

Dapat dilihat pada Tabel 6.2 bahwa hasil pengujian ukuran populasi untuk kloter siang menghasilkan rata-rata nilai *fitness* tertinggi sebesar 6,8658 yang mana terdapat pada ukuran populasi 90 dan rata-rata nilai *fitness* terendah yang dihasilkan adalah 6,78064 yaitu pada ukuran populasi 10. Hal ini disebabkan ukuran populasi yang semakin besar akan menghasilkan individu yang banyak. Semakin banyak individu maka semakin beragam variasi yang dihasilkan sehingga nilai *fitness* yang dihasilkan semakin besar dan semakin baik (Kusumaningsih, 2016).

Hasil pengujian ukuran populasi kloter siang pada Tabel 6.2 maka grafik hasil pengujian ukuran populasi untuk kloter siang dapat dilihat pada Gambar 6.2.



**Gambar 6.2 Grafik Hasil Pengujian Ukuran Populasi Kloter Siang**

Grafik Gambar 6.2 merupakan hasil pengujian ukuran populasi kloter pagi. Pengujian ukuran populasi dilakukan dengan cara menjumlahkan nilai *fitness* dari masing-masing populasi dimana nilai ukuran populasi merupakan nilai kelipatan yang dimulai dari kelipatan 10 sampai 100. Kemudian ditampilkan dalam bentuk grafik sehingga dapat diketahui nilai *fitness* tertinggi berada pada ukuran populasi ke-90 dengan nilai sebesar 6,8658.

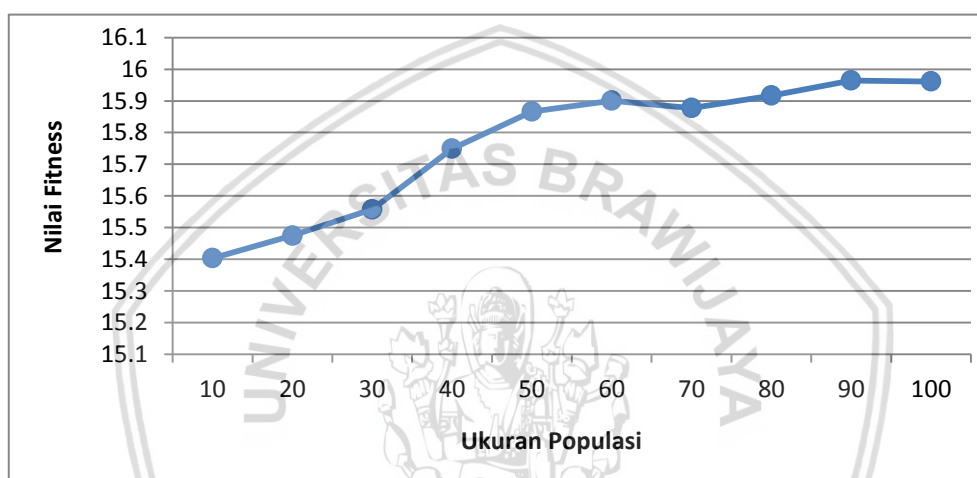
Selanjutnya untuk memudahkan penentuan ukuran populasi yang memiliki nilai *fitness* tertinggi pada kloter pagi dan siang maka dilakukan penggabungan rata-rata nilai *fitness* pada masing-masing ukuran populasi. Penggabungan rata-rata nilai *fitness* dilakukan dengan cara menjumlahkan nilai *fitness* kloter pagi dan kloter siang. Hasil penjumlahan pada pengujian ini dapat dilihat pada Tabel 6.3.

**Tabel 6.3 Penggabungan Hasil Pengujian Ukuran Populasi**

Jumlah Populasi	Penjumlahan <i>Fitness</i> Kloter Pagi dan Kloter Siang
10	15,40314
20	15,4747
30	15,5575
40	15,7491

50	15,8665
60	15,9009
70	15,8778
80	15,9171
90	15,9648
100	15,9618

Hasil penggabungan pengujian ukuran populasi masing-masing kloter pada Tabel 6.3 maka grafik penggabungan hasil pengujian ukuran populasi pada setiap kloter dapat dilihat pada Gambar 6.3.



**Gambar 6.3 Grafik Penggabungan Hasil Rata-rata *Fitness* Pengujian Ukuran Populasi**

Grafik Gambar 6.3 merupakan hasil penggabungan nilai *fitness* pada pengujian ukuran populasi. Penggabungan nilai *fitness* dilakukan dengan cara menjumlahkan nilai *fitness* pada kloter pagi dan kloter siang. Kemudian ditampilkan dalam bentuk grafik sehingga dapat diketahui nilai *fitness* tertinggi berada pada ukuran populasi ke-90 dengan nilai sebesar 15,9648 dan nilai *fitness* terendah berada pada ukuran populasi ke-10 dengan nilai sebesar 15,50314.

## 6.2 Pengujian Kombinasi Nilai *Crossover Rate* (Cr) dan *Mutation Rate* (Mr)

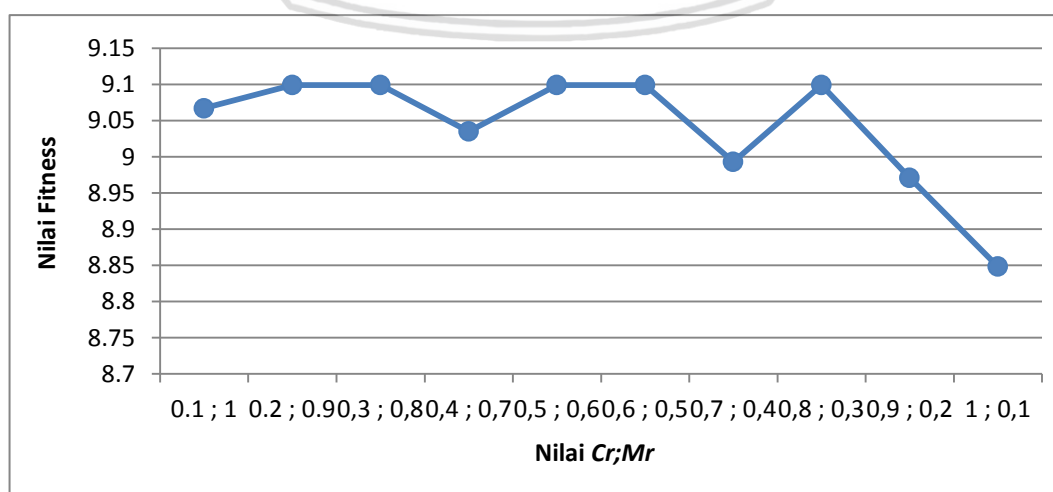
Pengujian kombinasi nilai *crossover rate* (*cr*) dan *mutation rate* (*mr*) dilakukan dengan kombinasi nilai *cr* dan *mr* dimulai dari 0,1 hingga 1 sebanyak 10 kali uji coba. Jumlah input data siswa yang digunakan pada pengujian ini sebanyak 10 data siswa untuk kloter pagi dan 12 data siswa untuk kloter siang. Hasil pengujian kombinasi nilai *crossover rate* (*cr*) dan *mutation rate* (*mr*) kloter pagi dapat dilihat pada Tabel 6.4 dan hasil selengkapnya dapat dilihat pada Lampiran C.2.

Tabel 6.4 Hasil Pengujian Kombinasi Nilai Cr dan Mr Kloter Pagi

Cr;Mr	Pengujian ke – i					Nilai Fitness
	1	2	....	9	10	
0,1 ; 1	9,0991	9,0991	....	9,0991	9,0991	9,0677
0,2 ; 0,9	9,0991	9,0991	....	9,0991	9,0991	9,0991
0,3 ; 0,8	9,0991	9,0991	....	9,0991	9,0991	9,0991
0,4 ; 0,7	9,0991	8,779	....	8,779	9,0991	9,0991
0,5 ; 0,6	9,0991	9,0991	....	9,0991	9,0991	9,0991
0,6 ; 0,5	9,0991	9,0991	....	9,0991	9,0991	9,0991
0,7 ; 0,4	9,0991	9,0991	....	9,0991	9,0991	8,9929
0,8 ; 0,3	9,0991	9,0991	....	9,0991	9,0991	9,0991
0,9 ; 0,2	9,0991	9,0991	....	9,0991	9,0991	8,9713
1 ; 0,1	9,0991	8,038	....	9,0991	8,779	8,8484

Pada Tabel 6.4 hasil pengujian kombinasi *cr* dan *mr* kloter pagi dapat dilihat bahwa rata-rata nilai *fitness* tertinggi yang didapatkan dari hasil kombinasi *cr* 0,8 dan *mr* 0,3 sebesar 9,0991 dan rata-rata nilai *fitness* terendah yang didapatkan dari hasil kombinasi *cr* 1 dan *mr* 0,1 sebesar 8,8484. Hal ini disebabkan nilai *crossover rate* yang terlalu rendah dan *mutation rate* yang terlalu tinggi menyebabkan metode algoritme genetika tidak dapat memperluas area pencarian, sedangkan jika nilai *crossover rate* yang terlalu tinggi dan *mutation rate* terlalu rendah menyebabkan algoritme genetika tidak dapat mencari ruang pencarian yang lebih efektif (Kusumaningsih, 2016).

Berdasarkan hasil pengujian kombinasi *crossover rate* (*cr*) dan *mutation rate* (*mr*) kloter pagi pada Tabel 6.4 maka grafik hasil pengujian kombinasi nilai *cr* dan *mr* kloter pagi dapat dilihat pada Gambar 6.4.



Gambar 6.4 Grafik Hasil Pengujian Kombinasi Cr dan Mr Kloter Pagi

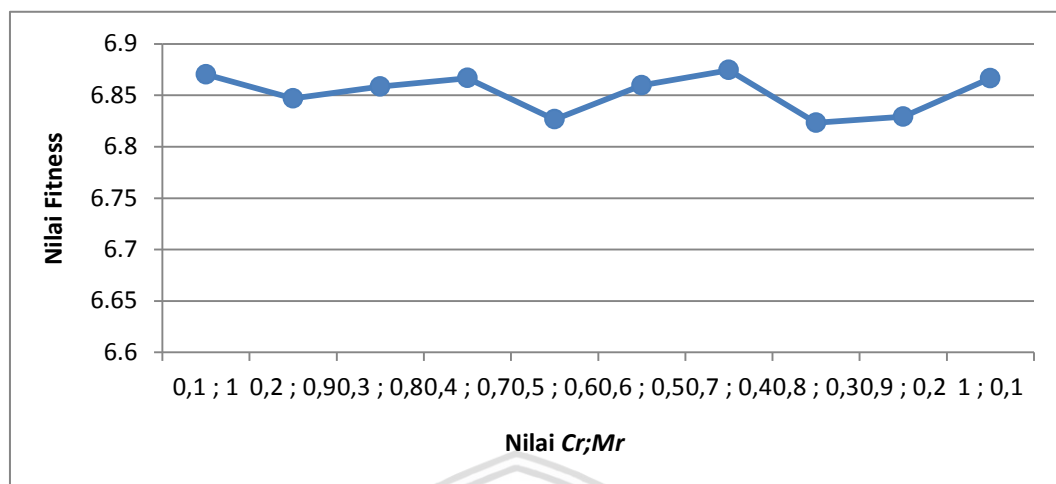
Grafik Gambar 6.4 merupakan hasil pengujian kombinasi *cr* dan *mr*. Pengujian kombinasi *cr* dan *mr* dilakukan dengan cara menjumlahkan nilai *fitness* dari masing-masing populasi dimana nilai *cr* dan *mr* merupakan nilai kelipatan yang dimulai dari kelipatan 0,1 sampai 1. Kemudian ditampilkan dalam bentuk grafik sehingga dapat diketahui nilai *fitness* tertinggi berada pada nilai *cr* 0,8 dan *mr* 0,3 dengan nilai sebesar 9,0991.

Selanjutnya hasil pengujian ukuran populasi pada kloter siang dapat dilihat pada Tabel 6.5 dan hasil selengkapnya dapat dilihat pada Lampiran C.2.

**Tabel 6.5 Hasil Pengujian Kombinasi Nilai *Cr* dan *Mr* Kloter Siang**

<i>Cr;Mr</i>	Pengujian ke – <i>i</i>					Nilai Fitness
	1	2	....	9	10	
0,1 ; 1	6,882	6,882	....	6,882	6,882	6,8703
0,2 ; 0,9	6,882	6,882	....	6,882	6,882	6,8469
0,3 ; 0,8	6,835	6,882	....	6,835	6,835	6,8585
0,4 ; 0,7	6,729	6,882	....	6,882	6,882	6,8667
0,5 ; 0,6	6,821	6,835	....	6,706	6,882	6,8266
0,6 ; 0,5	6,72	6,882	....	6,882	6,882	6,8597
0,7 ; 0,4	6,882	6,882	....	6,882	6,868	6,8745
0,8 ; 0,3	6,835	6,882	....	6,821	6,788	6,8233
0,9 ; 0,2	6,72	6,882	....	6,835	6,835	6,8292
1 ; 0,1	6,821	6,882	....	6,882	6,882	6,8665

Pada Tabel 6.5 hasil pengujian kombinasi *cr* dan *mr* kloter pagi dapat dilihat bahwa rata-rata nilai *fitness* tertinggi yang didapatkan dari hasil kombinasi *cr* 0,7 dan *mr* 0,4 sebesar 6,8745 dan rata-rata nilai *fitness* terendah yang didapatkan dari hasil kombinasi nilai *cr* 0,8 dan *mr* 0,3 sebesar 6,8233. Hal ini disebabkan nilai *crossover rate* yang terlalu rendah dan *mutation* yang terlalu tinggi menyebabkan metode algoritme genetika tidak dapat memperluas area pencarian, sedangkan jika nilai *crossover rate* yang terlalu tinggi dan *mutation rate* terlalu rendah menyebabkan algoritme genetika tidak dapat mencari ruang pencarian yang lebih efektif (Kusumaningsih, 2016).



**Gambar 6.5 Grafik Hasil Pengujian Kombinasi  $Cr$  dan  $Mr$  Kloter Siang**

Grafik diatas merupakan hasil pengujian kombinasi  $cr$  dan  $mr$ . Pengujian kombinasi  $cr$  dan  $mr$  dilakukan dengan cara menjumlahkan nilai *fitness* dari masing-masing populasi dimana nilai  $cr$  dan  $mr$  merupakan nilai kelipatan yang dimulai dari kelipatan 0,1 sampai 1. Kemudian ditampilkan dalam bentuk grafik sehingga dapat diketahui nilai *fitness* tertinggi berada pada nilai  $cr$  0,7 dan  $mr$  0,4 dengan nilai sebesar 6,8745.

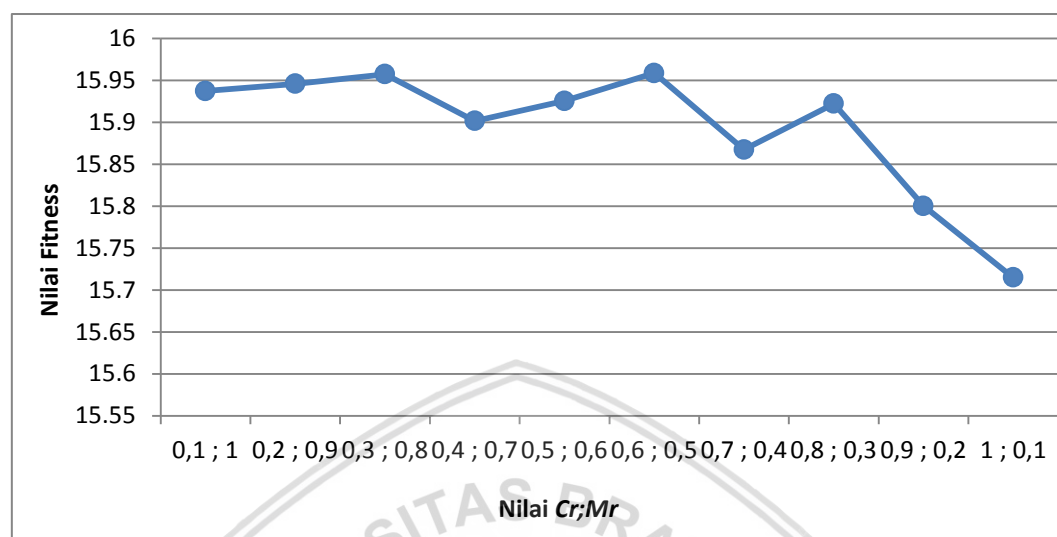
Selanjutnya untuk memudahkan penentuan kombinasi nilai  $cr$  dan  $mr$  pada kloter pagi dan kloter siang maka dilakukan penggabungan rata-rata nilai *fitness* pada masing-masing kloter. Penggabungan rata-rata nilai *fitness* dilakukan dengan cara menjumlahkan nilai *fitness* kloter pagi dan kloter siang. Hasil penjumlahan pada pengujian ini dapat dilihat pada Tabel 6.5.

**Tabel 6.6 Penggabungan Hasil Pengujian Kombinasi Nilai  $Cr$  dan  $Mr$**

$Cr;Mr$	Penjumlahan <i>Fitness</i> Kloter Pagi dan Kloter Siang
0,1 ; 1	15,9373
0,2 ; 0,9	15,9459
0,3 ; 0,8	15,9575
0,4 ; 0,7	15,9017
0,5 ; 0,6	15,9256
0,6 ; 0,5	15,9587
0,7 ; 0,4	15,8674
0,8 ; 0,3	15,9223
0,9 ; 0,2	15,8002
1 ; 0,1	15,7149



Hasil penggabungan pengujian kombinasi nilai *cr* dan *mr* masing-masing kloter pada Tabel 6.5 maka grafik penggabungan hasil pengujian kombinasi *cr* dan *mr* pada masing-masing kloter dapat dilihat pada Gambar 6.6.



Gambar 6.6 Grafik Penggabungan Rata-rata *Fitness* Kombinasi Nilai *Cr* dan *Mr*

### 6.3 Pengujian Konvergensi

Tujuan pengujian konvergen adalah untuk mendapatkan nilai *fitness* yang konsisten pada setiap generasi. Terdapat beberapa parameter yang digunakan pada pengujian konvergen kloter pagi yaitu jumlah *popsiz* sebanyak 90, generasi sebesar 1000, *cr* 0,6 dan *mr* 0,5. Hasil pengujian konvergen kloter pagi dapat dilihat pada Tabel 6.7.

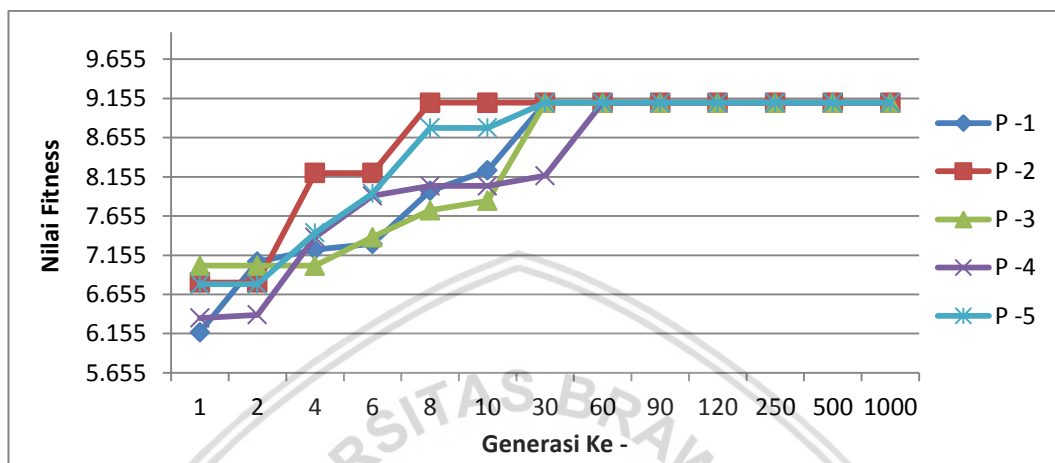
Tabel 6.7 Hasil Pengujian Konvergen Kloter Pagi

Generasi ke -	Percobaan Ke – <i>i</i>				
	1	2	3	4	5
	Fitness	Fitness	Fitness	Fitness	Fitness
1	6,1728	6,8073	7,0224	6,3532	6,7842
2	7,0721	6,8073	7,0224	6,3938	6,7842
4	7,2254	8,2034	7,0224	7,3855	7,4404
6	7,2992	8,2034	7,3855	7,9113	7,9428
8	7,9744	9,0991	7,7279	8,0385	8,7796
10	8,2372	9,0991	7,8492	8,0385	8,7796
30	9,0991	9,0991	9,0991	8,1699	9,0991
60	9,0991	9,0991	9,0991	9,0991	9,0991
90	9,0991	9,0991	9,0991	9,0991	9,0991
120	9,0991	9,0991	9,0991	9,0991	9,0991



250	9,0991	9,0991	9,0991	9,0991	9,0991
500	9,0991	9,0991	9,0991	9,0991	9,0991
1000	9,0991	9,0991	9,0991	9,0991	9,0991

Berdasarkan hasil pengujian konvergen kloter pagi pada Tabel 6.7 maka grafik hasil pengujian konvergen pada kloter pagi dapat dilihat pada Gambar 6.7.



**Gambar 6.7 Grafik Hasil Pengujian Konvergen Kloter Pagi**

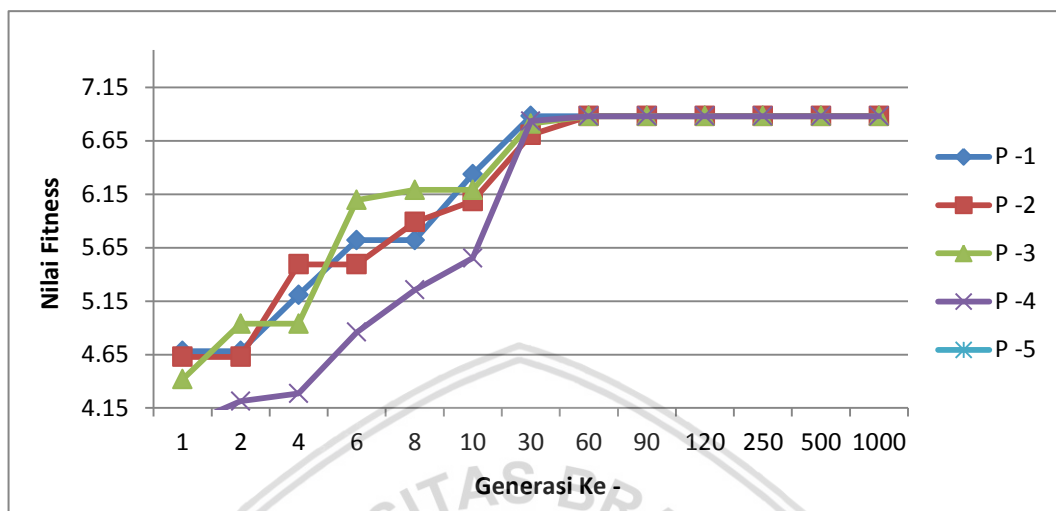
Gambar 6.7 merupakan grafik hasil pengujian konvergen kloter pagi. Hasil dari pengujian Gambar 6.7 menunjukkan bahwa nilai *fitness* mulai konvergen pada generasi ke-45 dengan rata-rata nilai *fitness* 9,0991.

**Tabel 6.8 Hasil Pengujian Konvergen Kloter Siang**

Generasi ke -	Percobaan Ke – i				
	1	2	3	4	5
	Fitness	Fitness	Fitness	Fitness	Fitness
1	4,683	4,629	4,418	3,976	4,231
2	4,683	4,629	4,938	4,214	4,366
4	5,208	5,494	4,938	4,286	4,878
6	5,72	5,494	6,097	4,859	5,154
8	5,72	5,889	6,191	5,254	5,39
10	6,337	6,086	6,191	5,552	5,753
30	6,882	6,706	6,811	6,835	6,811
60	6,882	6,882	6,882	6,882	6,882
90	6,882	6,882	6,882	6,882	6,882
120	6,882	6,882	6,882	6,882	6,882
250	6,882	6,882	6,882	6,882	6,882
500	6,882	6,882	6,882	6,882	6,882

1000	6,882	6,882	6,882	6,882	6,882
------	-------	-------	-------	-------	-------

Berdasarkan hasil pengujian konvergen kloter pagi pada Tabel 6.8 maka grafik hasil pengujian konvergen pada kloter pagi dapat dilihat pada Gambar 6.8.



**Gambar 6.8 Grafik Hasil Pengujian Konvergen Kloter Siang**

Gambar 6.8 merupakan grafik hasil pengujian konvergen kloter siang. Hasil dari pengujian diatas menunjukkan bahwa nilai *fitness* mulai konvergen pada generasi ke-45 dengan rata-rata nilai *fitness* 6,882. Sehingga dapat diketahui bahwa hasil solusi yang didapatkan dari sistem mulai stabil dan tidak mengalami perubahan sampai dengan generasi ke-45. Maka dari itu diketahui bahwa generasi ke-45 merupakan generasi yang optimal untuk mencapai konvergensi.

## 6.4 Hasil Analisis Global

Berdasarkan hasil pengujian populasi, pengujian kombinasi nilai *crossover rate* dan *mutation rate*, dan pengujian konvergen didapatkan nilai optimal parameter algoritme genetika pada kloter pagi sebagai berikut:

- Ukuran generasi : 1000
- Ukuran populasi : 90
- *Crossover rate* : 0,6
- *Mutation rate* : 0,5

Berdasarkan parameter diatas maka dapat dilakukan analisis global untuk melakukan pengujian data sampel aktual dari pengantaran siswa sekolah MI Salafiyah Kasim. Data sampel yang digunakan ada 3 hari yang dapat dilihat pada Tabel 6.9 dan data aktual yang lengkap dapat dilihat di Lampiran C.3.

Tabel 6.9 Data Aktual Rute Pengantaran

Hari	Data Aktual Urutan Pengantaran					
	Kloter Pagi			Kloter Siang		
	ID	Nama	Total Jarak (Km)	ID	Nama	Total Jarak (Km)
	S	Sekolah	22,6	S	Sekolah	34,00
	19	atiq		13	indah	
	14	alissa		14	hanum	
	18	firli		11	najib	
	15	melissa		12	syifa'i	
	17	arin		19	ririn	
	11	arum		10	asfira	
	13	syafa		9	fauzi	
	9	sekar		8	lila	
	20	khansa		7	shabrina	
	10	lutfi		24	abimanyu	
	2	ilham		20	ulfa	
	3	zaki		23	naura	
	4	ghozali		1	zahroh	
	5	zahra		2	naim	
	6	iza		22	ridwan	
	7	intan		4	nadzir	
	8	caca		5	dzikri	
	S	Sekolah		6	hasya	
				21	adib	
				17	rahma	
				S	Sekolah	
2	S	Sekolah	22,5	S	Sekolah	38,8
	....	.....		....	.....	
	S	Sekolah		S	Sekolah	
	S	Sekolah	27,8	S	Sekolah	47,3
	....	.....		....	.....	
	S	Sekolah		S	Sekolah	

Berdasarkan Tabel 6.9 maka pengujian sistem dilakukan dengan menggunakan parameter dan data di dalam tabel tersebut. Keluaran yang dihasilkan dari sistem dapat dilihat pada Tabel 6.10 dan data lengkap dapat dilihat pada Lampiran C.4.

**Tabel 6.10 Data Rekomendasi Hasil Sistem**

Hari	Data Aktual Urutan Pengantaran					
	ID	Nama	Total Jarak (Km)	ID	Nama	Total Jarak (Km)
1	S	Sekolah	21,97	S	Sekolah	29,5
	19	atiq		2	ilham	
	14	alissa		3	zaki	
	18	firli		4	ghozali	
	15	melissa		5	zahra	
	17	arin		6	iza	
	11	arum		7	intan	
	13	syafa		8	caca	
	9	sekar		9	sekar	
	20	khansa		13	syafa	
	10	lutfi		14	alissa	
	2	ilham		15	melissa	
	3	zaki		16	ahmad	
	4	ghozali		17	arin	
	5	zahra		18	firli	
	6	iza		19	atiq	
	7	intan		S	Sekolah	
	8	caca				
	S	Sekolah				
2	S	Sekolah	20,1	S	Sekolah	24,9
	....	.....		....	.....	
	S	Sekolah		S	Sekolah	
3	S	Sekolah	22,3	S	Sekolah	30,13
	....	.....		....	.....	
	S	Sekolah		S	Sekolah	

Pada Tabel 6.10 merupakan tabel data hasil rekomendasi sistem. Hasil dari rekomendasi tabel diatas akan melakukan perbandingan dengan data aktual, selanjutnya akan menghitung selisih dari jarak yang sudah didapatkan. Hasil selisih dapat dilihat pada Tabel 6.11.

**Tabel 6.11 Hasil Selisih**

Hari	Data Aktual Urutan Pengantaran					
	Kloter Pagi			Kloter Siang		
	Jarak Aktual (Km)	Jarak Sistem (Km)	Selisih	Jarak Aktual (Km)	Jarak Sistem (Km)	Selisih
1	22,6	21,97	0,63 (2,88%)	34,00	29,55	4,45 (13,08%)
2	22,5	20,1	2,4 (10,66%)	38,8	23,05	13,9 (35,82%)
3	27,8	22,3	5,5 (19,78%)	47,3	30,13	17,17 (36,30%)

Tabel 6.11 merupakan tabel hasil selisih pada dua kloter yaitu selisih jarak aktual dengan jarak sistem pada kloter pagi dan selisih jarak aktual dengan jarak sistem pada kloter siang. Berikut perhitungan untuk mencari selisih antara jarak aktual dan jarak sistem dihitung dengan cara hasil selisih dibagi dengan jarak aktual kemudian dikali 100%. Berdasarkan hasil perhitungan dengan rumus tersebut maka Tabel 6.11 menunjukkan bahwa sistem telah berhasil menghasilkan optimasi yang lebih baik dibandingkan dengan data aktualnya. Dapat disimpulkan bahwa sistem yang dibangun pada penelitian ini sudah berjalan dengan baik dan memberikan hasil yang optimal.

## BAB 7 PENUTUP

### 7.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan mengenai optimasi rute terpendek pada angkutan sekolah maka didapatkan pula kesimpulan sebagai berikut:

1. Hasil pengujian ukuran populasi diketahui bahwa nilai *fitness* tertinggi pada kloter pagi dan kloter siang berada pada ukuran populasi ke – 90 dengan nilai *fitness* sebesar 15,9648.
2. Hasil pengujian kombinasi *Cr* dan *Mr* diketahui bahwa nilai *fitness* tertinggi pada kloter pagi dan kloter siang berada diantara nilai *Cr* 0,6 dan *Mr* 0,4 dengan nilai *fitness* sebesar 15,9587.
3. Hasil pengujian konvergen diketahui bahwa nilai *fitness* pada kloter pagi mulai konvergen di generasi ke-45 dengan nilai *fitness* 9,0991 dan pada kloter siang mulai konvergen di generasi ke-45 dengan nilai *fitness* 6,882.
4. Hasil analisis global menggunakan parameter hasil pengujian sebelumnya diketahui bahwa sistem memberikan hasil rekomendasi rute yang optimal dengan menunjukkan selisih perbandingan jarak rute pengantaran pada data aktual dan data sampel, yaitu pada kloter pagi sebesar 5,5 km (19,78%) dan 17,17 km (36,30%).

### 7.2 Saran

Untuk dapat memperbaiki dan mengembangkan penelitian selanjutnya berikut adalah beberapa saran yang harapannya dapat diimplementasi oleh peneliti selanjutnya:

1. Untuk mendapatkan hasil rekomendasi yang lebih baik maka peneliti selanjutnya dapat menambahkan parameter waktu dan kondisi jalan. Sehingga proses pengantaran dapat menjadi lebih efisien dan tepat waktu.
2. Mengembangkan sistem pada penelitian ini pada *platform* yang berbeda seperti *web based system* atau aplikasi mobile sehingga lebih mudah untuk diakses dan digunakan oleh *user*.
3. Untuk sistem yang dikembangkan dalam penelitian selanjutnya harapannya dapat di implementasi menggunakan metode yang berbeda sehingga dapat diketahui hasil perbandingan masing-masing metode.

## DAFTAR PUSTAKA

- Beraunews, 2017. *Data Indonesia dan Dunia, Korban Laka Lantas Usia Remaja Urutan Kedua Terbanyak*, s.l.: [Online]. Tersedia di: <http://www.beraunews.com/lipsus-real-life/liputan-khusus/3002-data-indonesia-dan-dunia-korban-laka-lantas-usia-remaja-urutan-kedua-terbanyak?showall=1&limitStart=> [Diakses pada tanggal 9 Januari 2018]..
- Daryanto, M. R., 2012. *Model Pembelajaran Inovatif*. Yogyakarta: Gava Media.
- Dwivedi, V., Chauhan, T., Saxena, S. & Agrawal, P., 2012. *Travelling Salesman Problem using Genetic*. India, IMS Engineering College, NH-24, Ghaziabad, U.P-201009.
- Erdiwansyah, Gani, T. A. & Away, Y., 2016. Hibridisasi Simulated dengan Algorithm Evolutionary dalam Penyelesaian Travelling Salesman Problem (TSP). *Journal Online Teknik Elektro*, pp. 1-5.
- Greco, F., 2008. *Travelling Salesman Problem*. Rijeka: In Tech.
- Hannawati, A., T. & E., 2002. Pencarian Rute Optimum Menggunakan Algoritma Genetika. *Jurnal Teknik Elektro*, pp. 78-83.
- Joni I.D.A.B., V. N., 2012. Penentuan Jarak Terpendek Pada Jalur Distribusi Barang Di Pulau Jawa Menggunakan Algoritme Genetika. *Journal Nasional Pendidikan Teknik Informatika*, pp. 2089-8673.
- Kariono, 2011. *Layanan transportasi sekolah untuk menekan tidak masuk dan terlambat ke sekolah bagi siswa*. s.l.:s.n.
- Krishnanda, R., 2017. Optimasi Penentuan Rute Terpendek Pengambilan Sampah menggunakan Multi Travelling Salesman Problem. *Journal Pengembangan Teknologi Informasi dan Ilmu Komputer*, pp. 2227-2234.
- Kusumaningsih, F. D., 2016. Penerapan Algoritme Genetika Pada Optimasi Susunan Bahan Makanan Untuk Pemenuhan Kebutuhan Gizi Keluarga. *Fakultas Ilmu Komputer Universitas Brawijaya*.
- M.Dorigo, T., 2004. *Ant Colony Optimization*. s.l.:s.n.
- Mahmudy, W. F., 2013. *Modul Evolution Algorithm*. Malang: Fakultas Ilmu Komputer.
- Mahmudy, W. F., 2015. *Modul Dasar Teori Evolution Algorithm*. Malang: Program Teknologi Informasi dan Ilmu Komputer.
- Munawar, A., 2005. *Dasar-dasar Teknik Transportasi*. Yogyakarta: Beta Offset.
- Potvin, J. Y., 1996. Genetic algorithms for the. *Annals of Operations Research*, pp. 339-370.
- Suprayogi, D. A., 2015. Penerapan Algoritma Genetika Travelling Salesman Problem with Time Window: Studi Kasus Antar Jemput Laundry. *Jurnal Buana Informatika*, pp. 121-130.